for Atari Computers

U.S.A. $3.00
CANADA $3.50

Antic™

The ATARI® Resource

FEBRUARY 1984 VOLUME 2, NUMBER 11

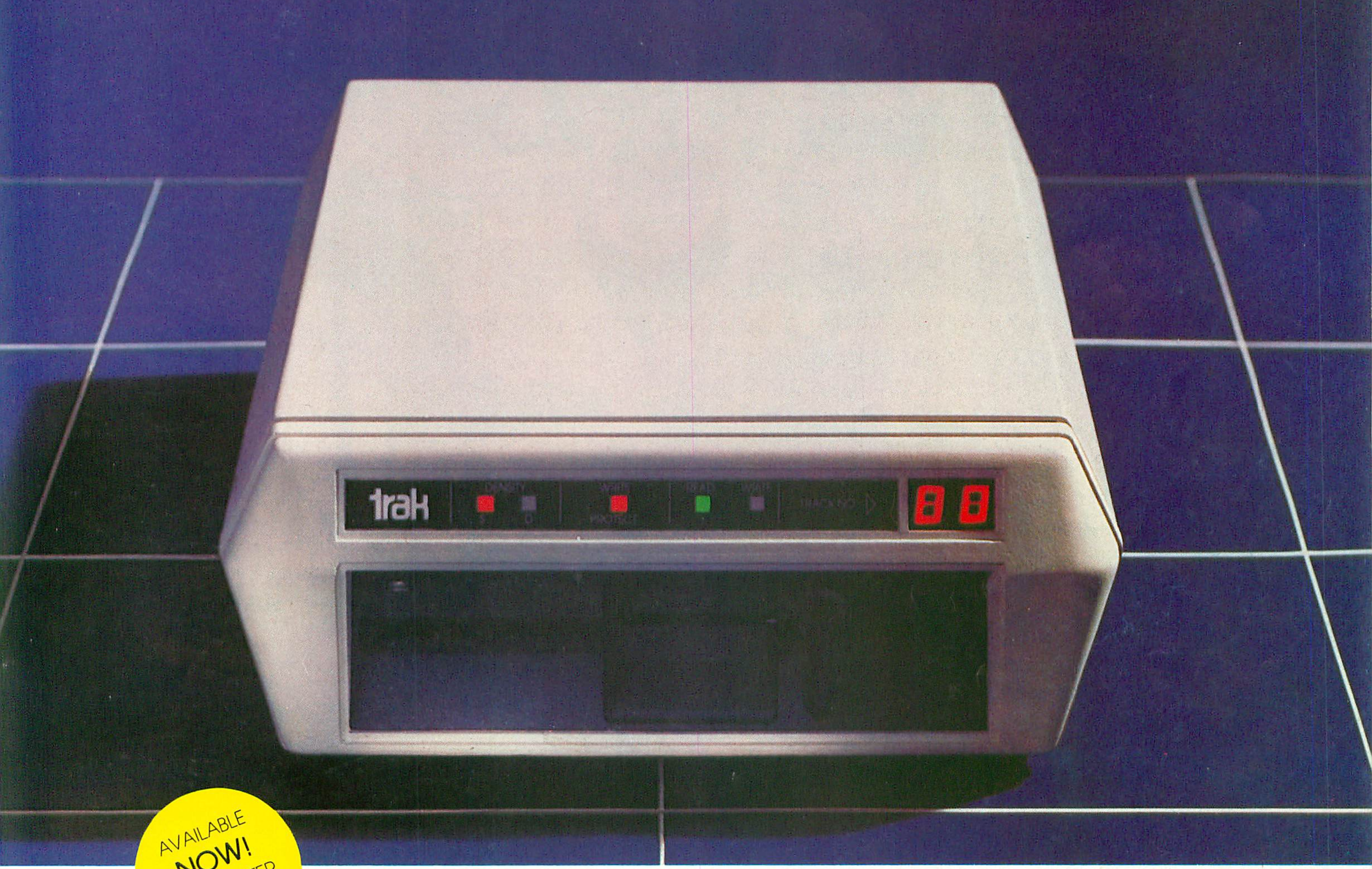
PERSONAL FINANCE—Programs & Products

- Atari Stock Portfolio
- Computer Combat Games
- Atari Joins the Olympics
- Education Wrap-up
- Books for Atari Logo
- 'Galactic Gloop' Game



0 74470 12728 02

WATCH OUT ATARI!
AT-D2 HAS LANDED!



AVAILABLE NOW! WITH PRINTER PORT



TRAK'S floppy and hard disk drive systems for the Apple and IBM-PC are available at many Computerland and other fine stores worldwide.

. . .WITH IT'S FLEET OF THE MOST ADVANCED SINGLE AND DOUBLE DENSITY DRIVE SYSTEMS IN THE UNIVERSE - TOTALLY COMPATIBLE WITH ATARI* COMPUTER MODELS 400, 800, 600XL, 1200XL, 1400XL, AND 1450XL.

DOCK YOUR FAVORITE PRINTER DIRECTLY TO OUR DRIVE'S BUILT-IN PARALLEL PRINTER INTERFACE. THERE'S NO NEED TO BUY AN EXPENSIVE ATARI 850 MODULE.

THE COMMAND CONTROL PANEL FEATURES READ, WRITE AND DENSITY INDICATORS, A TRACK COUNTER TO LOCATE EVERY BIT OF YOUR DATA AND A TOUCH-SENSITIVE SWITCH TO PROTECT IT.

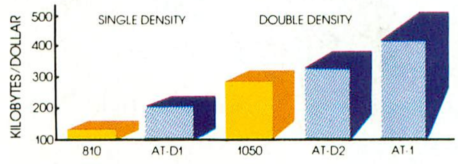
WARP SPEED OF UP TO 18 TIMES FASTER THAN **ANY** OTHER ATARI DRIVE IS ACHIEVED WITH OUR OPTIONAL TURBO-CHARGED SOFTWARE.

AWESOME POWER — DESTROY A PLANET OR LEARN HOW ONE IS CREATED. BALANCE YOUR CHECKBOOK OR WRITE A BESTSELLER. LEARN A NEW LANGUAGE OR CREATE ONE.

IT'S MAGNIFICENTLY SIMPLE! — JUST PLUG IT IN AND LIFT-OFF!

*Atari is a registered trademark of Atari, Inc.

TRAK GIVES YOU THE MOST STORAGE FOR YOUR DOLLAR!



OUR FLAGSHIP, AT-D2, WITH TWICE THE STORAGE CAPACITY OF THE ATARI 810 DRIVE PLUS A PRINTER INTERFACE AND COMMAND CONTROL PANEL, HAS A VALUE OF OVER $1000 YET COSTS LESS THAN $500!

SEE YOUR DEALER NOW FOR A TEST FLIGHT, OR

call toll free: **1-800-323-4853**
in Illinois, call collect: 1-312-968-1716
TWX 910-222-1848

trak™ A commitment to quality

microcomputer corporation 1511 Ogden Ave., Downers Grove, IL 60515

★★BONUS GAME★★

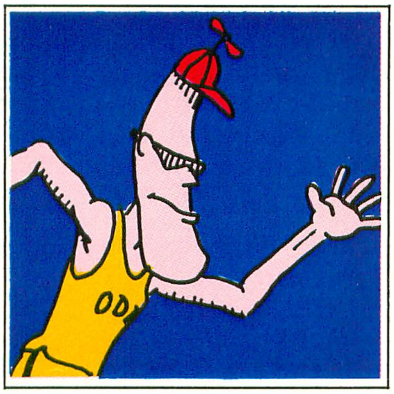
Antic

The ATARI Resource FEBRUARY 1984 VOLUME 2, NUMBER 11



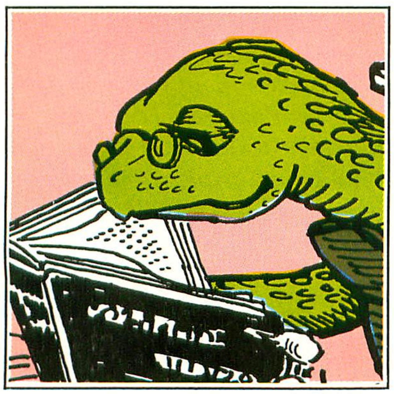
page 44

FEATURES



page 19

DEPARTMENTS



page 23

Antic
The ATARI Resource

Publisher
James Capparell
Managing Editor
Robert DeWitt
Assistant Editor
Christopher Rauber
Technical Copy Editor
David Duberman
Editorial Assistant
Caitlin Morgan
Editorial Clerk
Arla Ertz
Contributing Editors
Carl Evans
Ken Harms
John & Mary Harrison
David & Sandy Small
Jerry White
Art Director
Marni Tapscott
Production Supervisor
Kyle Bogertman
Contributing Illustrators
Beatrice Benjamin
Bill Boyer
John Musgrove
Cover
Marni Tapscott
Circulation Manager
Les Torok
Subscriptions
Julianna Hoffman
Shipping
Augustus Jones
Accounting
V.J. Briggs
Advertising/Production Coordinator
Linda Tapscott
Advertising Sales
Steve Randall

ANTIC is an independent periodical not affiliated in any way with Atari, Inc. ATARI is a trademark of Atari, Inc. All references to Atari products are trademarked and should be so noted.

February 1984
Volume 2, Number 11
ANTIC—The ATARI Resource, is published twelve times per year by ANTIC Publishing. Editorial offices are located at 524 Second Street, San Francisco, CA 94107. ISSN 0745-2527. Second Class Postage paid at San Francisco, California and additional mailing offices. POSTMASTER: Send address change to **ANTIC**, 524 Second Street, San Francisco, CA 94107.

Editorial submissions should include program listing on disk or cassette, and text file on media and paper if text was prepared with a word processor. Media will be returned if self-addressed stamped mailer is supplied. **ANTIC** assumes no responsibility for unsolicited editorial material.

General Offices (415) 957-0886
Advertising Sales (415) 661-3400
Credit Card Subscriptions
outside California (800) 227-1617 ext. 133
inside California (800) 772-3545 ext. 133

I/O BOARD

INTERFACE ALTERNATIVES

I have an Atari 800, and have recently purchased an Epson RX-80 printer. Can you suggest alternatives to the 850 (Atari) and the MPP-1100 (Microbits Peripherals) interfaces. I want to take into consideration future additions to the system. Also, which disk drives are the best choices for the Atari?

R. P. Stein
Audubon, PA

Axiom makes a $99 printer interface. We haven't evaluated it yet, but expect to shortly. Also, several disk drives have built-in printer interfaces. Two of these are the Percom (special model) and the Trak. We haven't yet evaluated the performance of these drives, but are planning to publish a disk drive survey in the near future. See "Mission Redux: Disk Drive Daze" (ANTIC, December 1983) for a survey of disk drives from the point of view of the professional programmer.

—ANTIC ED

COMPUTERIZED GRADING

In the October 1983 issue of ANTIC, the authors of "Flip Side" hint at the existence of programs that keep track of grades for a semester. Where can I find these programs?

Don Sward
Northfield, MN

We're aware of two commercial products that perform classroom record-keeping chores: Easy Grader Rev. 1.1 (APX) and Teacher's Gradebook (Dynacomp, Inc.).

—ANTIC ED

THE BEST YET!

Congratulations on your December issue! It's your best ever. I've been reading ANTIC since you started, and it's been great fun to watch you grow.

Jerry Dea
Secretary
Atari Anonymous—
A Users Group
Upland, CA

TYPO HELPER

ANTIC's debugging program, "TYPO," is an excellent tool, but it's a little slow. Most people have to run it several times to find all of their typos and, because it checks the entire program each time, a lot of time is wasted checking portions of the code that you already know are correct. But by changing line 32110 and adding line 32175 (see below), you can force TYPO to start checking at a line number that you specify.

This line number should match a beginning line number of one of the ranges in the Typo Table for the listing involved. Please note that these new lines add a variable to TYPO; as a result, the variable checksum provided by this modified version of TYPO will not match the one printed in the magazine. Therefore, you should use the original version of TYPO until the two variable checksums are identical. Then you can use ENTER to merge these modified lines into the program:

```
32110 INPUT Q$:? "Starting Line
      Number";:INPUT STLINE:
      OPEN #QF,12,0,Q$:
      QREM=0
32175 IF QLINE<STLINE THEN
      QADDR=QADDR+PEEK
      (QADDR+2):G.32170
```

Ron Bishop
Owasso, OK

Thanks for the useful tip. By the way, a revised version of Bill Wilkinson's article on TYPO ("TYPO Revisited, Again") appears in this issue for the benefit of our many new readers. It explains how to use TYPO to check other program listings published in ANTIC and to check itself.

—ANTIC ED

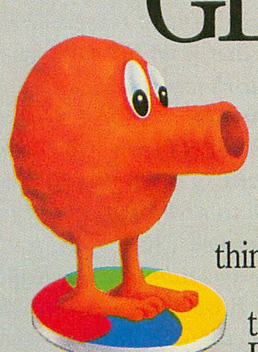
FOND OF OUR FONT

I'd like to compliment you on your monospaced program listings. Together

continued on page 8



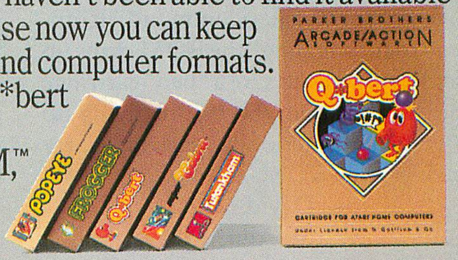
HOW TO GET Q*BERT™ OUT OF YOUR SYSTEM.



If you've been wanting to play Q*bert, but haven't been able to find it available for your home system, your time has come. Because now you can keep things hopping with any of these popular home video and computer formats.

Get going to your nearest video store and get Q*bert today. And while you're there, check out Parker Brothers' POPEYE,® FROGGER,™ TUTANKHAM,™ and SUPER COBRA.™ All the great Arcade Action® games, now in all the great home formats.

PARKER BROTHERS



with your customized font and "TYPO," they make ANTIC by far the easiest magazine to copy programs from. Thanks for a great magazine.

Brian Weiss
Pierre, SD

ROCKING ROMS

I've always been convinced that ROM cartridges are more rugged than cassette or disk media, but I was recently surprised to find that my **Missile Command** and **Star Raiders** cartridges wouldn't work. Missile Command put white squares on the screen, and Star Raiders generated only the familiar blue Memo-Pad. And, of course, the 90-day warranties for the games had long since expired.

Since I'm an electronics technician, I decided to do some experimenting. I started with Star Raiders, and tried pressing [RESET] and re-inserting the cartridge several times. No luck. Then I realized that if the Memo-Pad was coming up, the computer must think there was *not* a cartridge in the left slot. Since Star Raiders *was* plugged into the slot, I realized that something was not making proper contact.

I removed the cartridge's metal lid and looked inside. Everything looked okay, so I pried each of the two ROM chips a little and then carefully pushed them back into their sockets. Then I reassembled the cartridge and headed back to my 800.

I plugged the cartridge back in, and was greeted by the slow drone of the engines at Warp Three, and a front view of the galaxy with no Klingons in sight. Success! Moments later, the same fix worked for Missile Command. It appears that "wiggling" the ROM chips up and down a bit can save a "bad" cartridge from the trash bin. So before giving up on your malfunctioning cartridges, try rocking your ROMs.

David Mundy
St. Ann, MO

AEROBIC COMPUTING

I've been using the Suncom Aerobic Joystick with my Atari 800, and it does a great deal to relieve the tedium of my half-hour exercise routine every morning. Galaxian has provided me with a lot of entertainment during these sessions, but I'd like to try some other games for variety's sake. Can you suggest any games that take advantage of this joystick's special capabilities? The most important thing to remember is that the faster you pedal, the faster the fire button shoots. Also, you can't aim very precisely when you're pedalling away at 20 miles per hour.

Helen Phillips
Arvada, CO

The Aerobic Joystick should work best with games that require rapid and continual pressing of the fire button. This type of game is also ideal for testing joysticks and modules that add a "rapid-fire" capability to your system. Zaxxon (DataSoft) and Defender (Atari) both fit the bill nicely and are terrific games as well. —ANTIC ED

IN THE DATA FILE DUMPS

I'm trying to learn data file programming using Atari Microsoft BASIC. In an earlier letter, I asked if there was a way to produce a program that allows you to use a user-defined variable for the file name in the OPEN #iocb,"D:filename" file-access command. The book I was using at the time described how to do this in Microsoft-80, but the OPEN command is entirely different. You advised me to use a string variable in place of the file specification in the OPEN command. Unfortunately, I don't know what you mean by the term "file specification." Would it be possible for you to give me a specific example of the correct format? I'd also like to be able to specify what type of file to open.

Jerry Steinberg
Brooklyn, NY

If you want the user to input only the file name, you must insert the device specification (i.e., D:) in front of the fiilename. This is done via string manipulation.

The best solution we've found is to include a routine that checks for the device specification. If it is not present, the routine adds it.

Once you've derived the proper file specification (including device specification and filename), use the string variable, without quotes, as follows:

20 OPEN #1,N$OUTPUT

For further information on string manipulation, see Chapter 10 of your Microsoft BASIC manual.

—ANTIC ED

KOALA IS COMPATIBLE

I recently purchased the Koalapad for my Atari, and I'm very impressed by it. However, I'm dismayed by the fact that its picture-storing method is incompatible with other graphics programs, such as DataSoft's Micro-Painter. Do you know of a remedy?

Alek Grguric
Don Mills, Ont., Canada

You're in luck! You can store pictures that were created with Koalapad's Micro Illustrator program in a standardized format which is compatible with any program that can load a Graphics 8 or Mode E (Micro-Painter) screen. When you want to save your picture, go to Micro Illustrator's icon menu and position the cursor over the disk symbol. Then press [INSERT] on your computer's keyboard. This will save your current picture to disk in standard format as a file named PICTURE.

This remedy was submitted by Robert H. Watson of Brooklyn, NY. He also noted that you can load any file named PICTURE by moving the cursor to the disk symbol and pressing [CLEAR],

continued on page 10

I/O BOARD

but this didn't work when we tried it. If you load a picture created with Micro Illustrator into Micro-Painter, you'll get an error message and the picture's colors will not look right, because the four appended color bytes that Micro-Painter expects are not present. However, the entire picture can be loaded, and its colors can be adjusted. —ANTIC ED

MINDING THE MODEM

I recently purchased a 1200-baud Hayes Smart Modem. What do I need to connect it to my Atari 800 and run it? Please bear in mind that I'm a babe in the woods.

Milton Yuan
Rutherford, NJ

You'll need an Atari 850 interface to connect your Smart Modem to your Atari; TELETALK, by DataSoft, is one of several programs that will allow you to control the modem. For more information on the subject, see "Atari Terminal" (ANTIC, May 1983, page 32) and watch for our May issue this year, which will focus on the latest developments in the field of communications.

—ANTIC ED

HELP!

GTIA-SKETCHPAD BUGS

I've found a couple of minor bugs in "GTIA Sketchpad" (ANTIC, December 1983). If you move the cursor to the top of the screen in the "Doodle" mode while you're pressing the joystick's fire button (to create a wide line), you get an error message. To remedy this, change line 800 as follows:

```
800 IF YP<N1 THEN YP=148
```

If you move the cursor to the far left of the screen while pressing the fire button, it doesn't wrap around. To fix this, make the following changes to line 790:

```
790 IF XP<N1 THEN XP=78
```

These changes should make debugging easier.

Will Cronenwett
Norman, OK

TAKE IT FROM THE TOP

There is a minor flaw in "Air Raid 2000" (ANTIC, November 1983) by Erik Wolpaw. The title of the game flashes on the screen constantly. This is distracting and hard on the eyes. To remedy this, simply change the inverse-video title of the game in lines 1120 and 1370 to normal video.

All in all, however, I'd like to congratulate Erik on a job well done — keep it up!

Bill Mason
Redding, CA

WRONG ADDRESS

In the "ANTIC Pix Controllers" survey (December 1983), we gave an incorrect address for Questar Controls, Inc., the manufacturer of the Questar Control Console and the Blaster. Their correct address is: 799 Main St., Half Moon Bay, CA 94019.

A LOT OF ROT

The listing for my new "DOS Sectors to Forth" article (ANTIC, page 32, October 1983) contains a misprint at the end of the fourth line. The word "tROT" should be "<ROT." This command causes a left rotation of the stack — in other words, it moves the top of the stack to the third position from the top. This word can be implemented as "LROT" in some versions, and is defined as:

```
: <ROT ( N1 N2 N3 — N3 N1 N2 )
      ROT ROT ;
```

BIll Van Hassel
New Hope, PA

EPSON RX-80 PRICE

We provided incorrect price information on the Epson RX-80 printer in our recent "Printer Survey" (ANTIC, January 1983). The correct prices are $399 for the standard model and $599 for the RX-80 with the friction feature. The $299 price was an authorized introductory sale price.

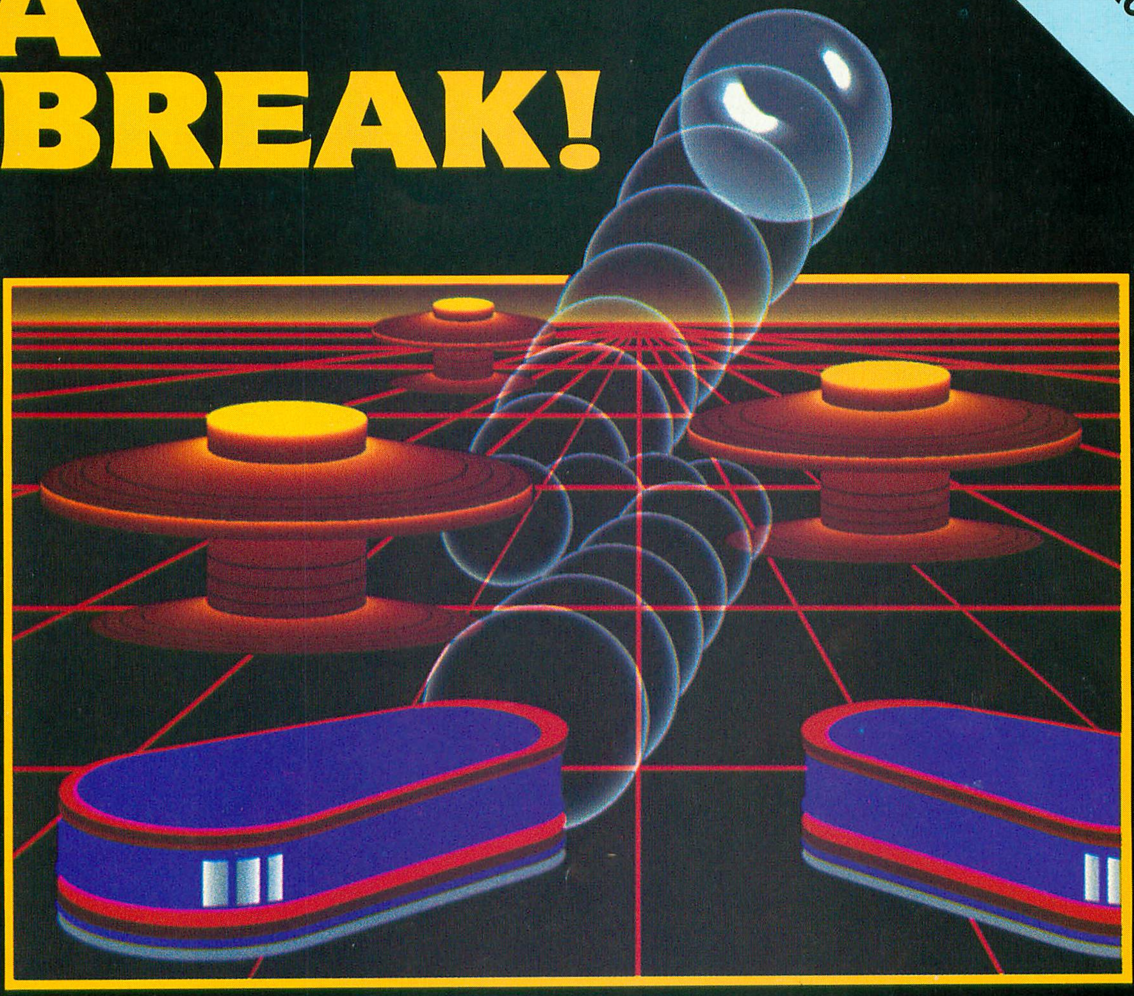
TAG

In the game "Tag" (ANTIC, November 1983, p. 84), the screen is supposed to clear after every minute of play. However, due to an oversight, it does not. To effect this feature, change line 95 to:

```
95 POKE 540,60:S=S+1:IF S=60
   THEN M=M+1:?#6;CHR$(125):
   S=0
```

For Atari computers with 32K or more memory

TAKE A BREAK!



WITH NIGHT MISSION
PINBALL

You deserve the best. You've earned it. Now reward yourself with a session of **Night Mission PINBALL**, the most realistic and challenging arcade simulation ever conceived! ■ Stunning graphics and dazzling sound effects put **Night Mission PINBALL** in a class by itself. Game features: multi-ball and multi-player capabilities, ten different professionally designed levels of play, and an editor that lets you create *your own* custom modes. ■ So take a break with **Night Mission PINBALL** from SubLOGIC. Winner of *Electronic Games* magazine's 1983 Arcade Award for Best Computer Audio/Visual Effects.



See your dealer . . .

or write or call for more information. For direct orders please add $1.50 for shipping and specify UPS or first class mail delivery. Illinois residents add 5% sales tax. American Express, Diner's Club, MasterCard, and Visa accepted.

Order Line: 800/637-4983

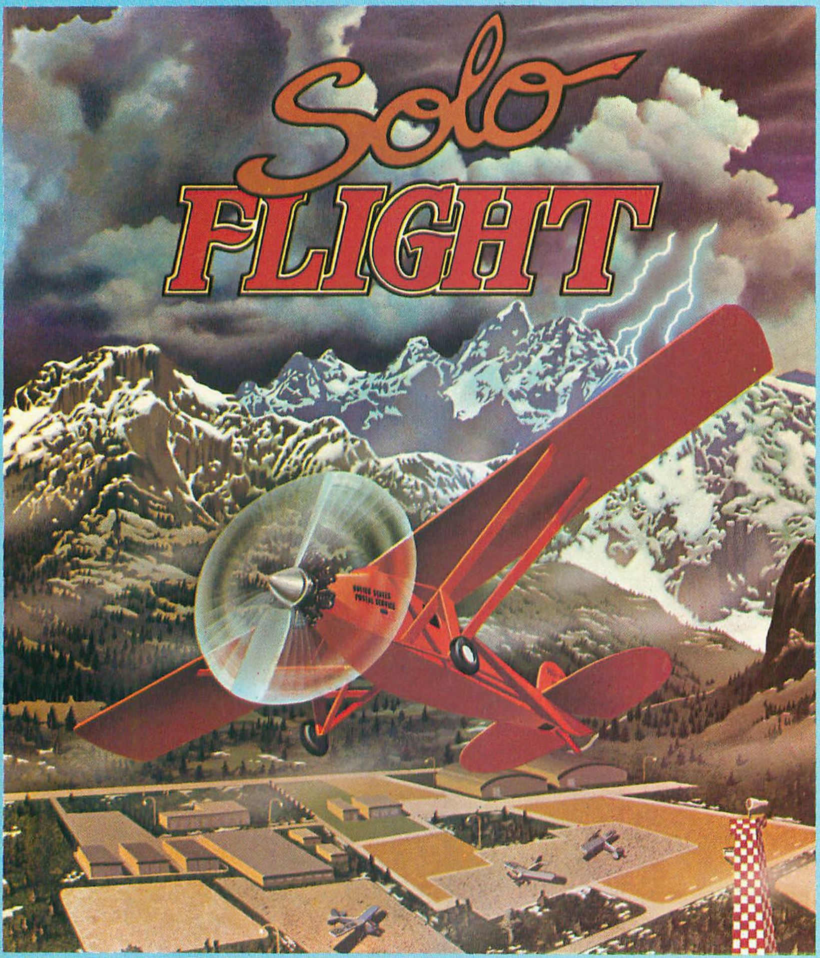
subLOGIC
Corporation
713 Edgebrook Drive
Champaign IL 61820
(217) 359-8482 Telex: 206995

AT LAST!!! A REAL FLIGHT SIMULATOR!!!

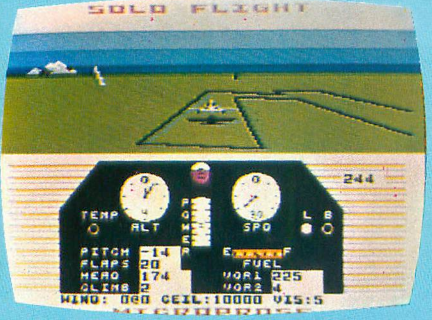
— From The Creators of HELLCAT ACE!

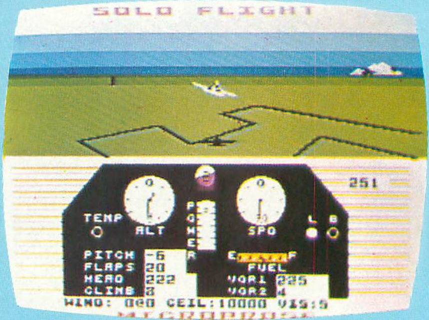
A real flight simulator allows you the pilot to accomplish Takeoffs, Landings, Cross-Country Navigation, and Emergency Procedure practice under Clear (VFR), Cloudy (IFR), and Cross-Wind conditions. SOLO FLIGHT is such an advanced simulation that uses realistic three dimensional terrain graphics, actual configuration instrument panel, multiple air navigation maps, and accurate aircraft performance characteristics to provide the challenge, thrill, excitement, and joy of the flight experience!!!

- Extensive flight manual including Basic Flying, Instrument Flying, Maps, and Charts.
- Three geographical air navigation areas each with multiple airports and multiple runways.
- Dual Radio Navigation (VOR) and Instrument Landing System (ILS).
- On-screen flight path review between flights.
- Multiple Landing Approach Charts and Design-your-own approach feature.



- Takeoffs and Landings under all weather conditions.
- Cross-Country Navigation via Landmarks, VOR Stations and Dead Reckoning.
- Emergency Procedure Practice.
- Air Mail delivery game to test your flying skills.
- Landing ratings for smoothness and speed.
- For all Atari computers, Commodore 64, Disk and Cassette . . . $34.95.





Look for MicroProse's two new exciting real time combat simulations — MIG ALLEY ACE & NATO COMMANDER.
Write or call for our Free Catalog.

If you cannot find our games at your local store, you can order by MasterCard or VISA, Money Order, COD or Check. Add $2.50 for Postage and Handling. Maryland Residents add 5% Sales Tax.

MicroProse Software

DEALER INQUIRIES INVITED.

10616 Beaver Dam Road, Hunt Valley, MD 21030
(301) 667-1151

IT'S OFFICIAL!

Atari joins the U.S. Olympic Team

by DAVID F. BARRY

The 1984 Summer Olympics have not yet opened in Los Angeles, but Atari is already having a major impact on the Olympic effort. As befits a company of Atari's stature, it is sponsoring the U.S. Women's Volleyball Team, which is considered to be one of the top contenders for the gold medal in this Olympic sport. This sponsorship is only one of several agreements Atari has entered into with the Los Angeles Olympic Organizing Committee (L.A.O.O.C.) for the 1984 Summer Games.

Atari has also contracted with the L.A.O.O.C. to be the Summer Games' sole sponsor of home computers, arcade games and home video games. This contract gives Atari the right to use the Olympic symbol, language, and logo on any of its promotional materials.

In addition, Atari has contracted with ABC Sports to air 25 commercials during the Winter Olympics in in Sarajevo, Yugoslavia (February 8–19), and 60 commercials during the Summer Olympics in Los Angeles (July 28 to August 12). Consequently, it should be difficult to watch the 1984 Olympics on television without being aware that Atari is on the scene!

Atari is also donating approximately $75,000 worth of equipment to be used in conjunction with the Summer Games.

David F. Barry is a technical writer in the computer field, and the author of an upcoming book on the word-processing program Wordstar.



Thus far, Atari has arranged to provide computer games for use by athletes in the Olympic Village, and to set up arcade games at the ABC International Broadcast center for use by some 2,000 media representatives from around the world.

To further dramatize its commitment to the U.S. Women's Volleyball team, Atari is sponsoring the team's current cross-country tour, which pits the women against some of America's top-ranked, men's collegiate varsity teams. In support of this commitment on the part of Atari, team members will be wearing Atari patches on their uniforms.

continued on next page

The tremendous strides that the Women's Volleyball Team has made in recent years are due in large part to its computerized training camp. At the Coto Research Center in Coto de Caza, California (70 miles south of Los Angeles), computers are providing coaches and trainers with such detailed images of their athletes' performance that the word "guesswork" is quickly disappearing from the Olympic vocabulary. Five years ago, the Women's Volleyball Team was unranked in world competition. Today, it is an internationally recognized contender for the Olympic gold medal.

The computerized facilities in Coto de Caza operate under the guidance of Dr. Gideon Ariel, a pioneer and renowned leader in the field of biomechanics, the computerized study of human movement.

Dr. Ariel uses high-speed cameras to film the movements of athletes. The frames of film are then projected onto a special screen, traced with a magnetic pen, fed into a computer for analysis, and then output as "three-dimensional" stick figures. The result is a highly-precise, graphic image of the physical movement involved. From these images, Dr. Ariel can determine such elusive variables as the optimal angle at which to spike a volleyball, or the most efficient way to make a lateral move on the volleyball court. He does this by comparing the graphic image of the actual motion with a hypothetical "optimum" motion developed in the laboratory. His aim, which fits in well with the ideals of the Olympics themselves, is to bring the actual closer to the optimum.

Dr. Ariel has also used biomedical analysis to study opposing volleyball teams. In the process, he has dissected their movements, studied their strengths and weaknesses, and provided fresh new insights about their styles of play to the rapidly improving U.S. Women's Team.

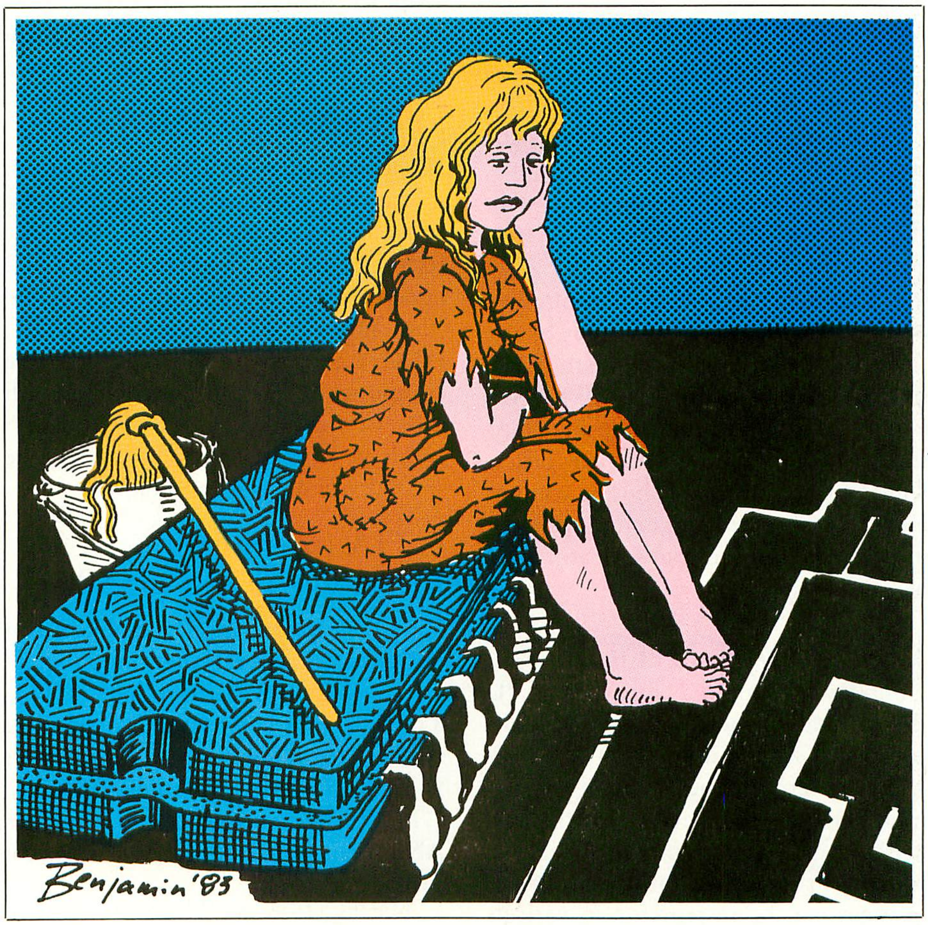
Stay tuned for more news about Atari's involvement in the 1984 Olympics in a future issue of ANTIC. In the meantime, you may begin to find yourself looking at your Atari in a different light. After all, it's now the Official Home Computer of the 1984 Olympics.

A CINDERELLA STORY

The half-life of a computer drudge

by FRED PINHO

Most computer owners know that a computer needs a "language," such as BASIC, to function. These languages are the "belles of the ball." But, as in the fairy tale, there is also an unsung drudge in the computer that these languages need to be able to do their thing. This hard-working servant goes by various names: Master Control Program, Operating System, System Monitor, etc. Atari chose to call their version simply, the Operating System (OS).



WHAT IS THE OS?

The OS is a gut-level program that manages the resources of the computer and controls communications with the computer's peripherals. A peripheral is a hardware device such as a keyboard, television, joystick, cassette drive, disk drive, or printer.

At the heart of a personal computer is the microprocessor, often referred to as the Central Processing Unit (CPU). In the Atari computers, the 6502 chip is the CPU. While a BASIC program is interacting with this chip, there are many housekeeping activities that must be monitored and controlled. These tasks range from controlling the TV display to establishing communications with a cassette recorder. If the programmer had to keep track of all of these activities as well as his or her program, programming would be very tedious. Thus, operating systems evolved to take some of this burden away from the programmer.

Fred Pinho is a biochemical research engineer and a self-taught programmer who is interested in BASIC and assembly language. The Atari 800 is his first computer.

Any operating system is a group of machine-language programs. Machine language is the microprocessor's mother (and only) tongue. The Atari's 6502 chip is an 8-bit microprocessor. This means that it handles and processes eight bits

continued on next page

of information (one byte) at a time. These eight pieces of information are expressed in binary notation. That is, the processor is only concerned about whether a bit is on (set) or off. There are no intermediate values to consider. Setting these bits in various combinations causes various circuits within the chip to be opened or closed. For our purposes, a byte expresses numbers in binary notation as follows:

Bit number — Decimal value if bit is set — Value as a power of 2

7	6	5	4	3	2	1	0
128	64	32	16	8	4	2	1
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

Note that each bit doubles in value as we go from right to left. We are, in effect, counting in powers of two. If a bit is not set, the value of that bit is zero. Thus, to determine the value of a byte, add up the values of all the bits that are set. Remember, position counts.

With this system, the highest number that can be represented in eight bits is 255, if all bits are set. But note that 256 numbers can be expressed. All bits off equals zero, and to the computer, zero is as important a number as any other. This may sound trivial, but as you get into more advanced programming, keeping this fact in mind can save you a lot of debugging time.

WHAT CAN IT DO?

Contrary to popular lore, the CPU doesn't really "know" numbers (binary or otherwise). All it knows is that voltages on certain lines cause predetermined changes to take place in other circuits. Chip designers have taken care to see that the results of these circuit changes can be interpreted in various ways. Once a value is in the accumulator, many other operations are possible. For example, if you send the value 169 to the 6502 chip, followed by the value 10, the chip will "load" 10 into a part of itself called the "accumulator."

The chip's repertoire is extremely limited. Basically, it can add, subtract, compare two numbers, perform simple versions of IF/THEN and GOSUB commands, and move data around in memory. These talents certainly won't qualify the 6502 for a stage career. Yet, from its humble (but *very* fast) skills, all the wonderful graphics and computational capabilities of today's software have evolved.

Physically the Atari OS is contained in several integrated circuit (IC) chips that are separate from the CPU. In the Atari 800, the OS is easily removable. It resides on the first large cartridge (sometimes called a board) in the memory bay under the removable top. Ranked behind it are the RAM cartridges. The OS is stored in fixed position in memory (locations 55296–65535); this position is located at the very top of the available memory space.

SPECIALTY CHIPS

What are the resources that the OS must manage? In addition to the microprocessor chip and RAM memory, there are four specialized chips that aid the 6502 in performing its job. These are ANTIC (that name sounds familiar!), GTIA, POKEY and PIA. The workings of these components must be integrated with those of the 6502, a job at which the OS excels. A short description of each of the specialty chips is given below:

GTIA (memory locations 53248–53505)

This Television Interface Chip was designed to process the video signal. Early versions of the Atari had a CTIA. The difference between the two chips is that GTIA supports Graphics Modes 9–11, while CTIA does not. Basically, GTIA converts digital commands from ANTIC into a video signal that is sent to the television.

The data from ANTIC contains information for the Graphics Mode desired. While processing this data, GTIA adds its own information, which includes color values and Player/Missile graphics. GTIA also provides collision detection. (Collision for the Player/Missile system is defined as two objects overlapping on the screen.) Finally, this busy little chip monitors the console keys.

ANTIC (locations 54272–54783)

This chip is actually a specialized microprocessor in its own right. It controls the screen display through instructions to C/GTIA. ANTIC has its own mini-program called the *display list*. Upon a graphics call from BASIC, the Operating System sets up the display list at a specific spot in RAM. The display list then tells ANTIC exactly how to set up the TV screen and where to find the data to be displayed. ANTIC retrieves the display data and instructions from the display list and translates them into instructions for C/GTIA.

POKEY (locations 53760–54015)

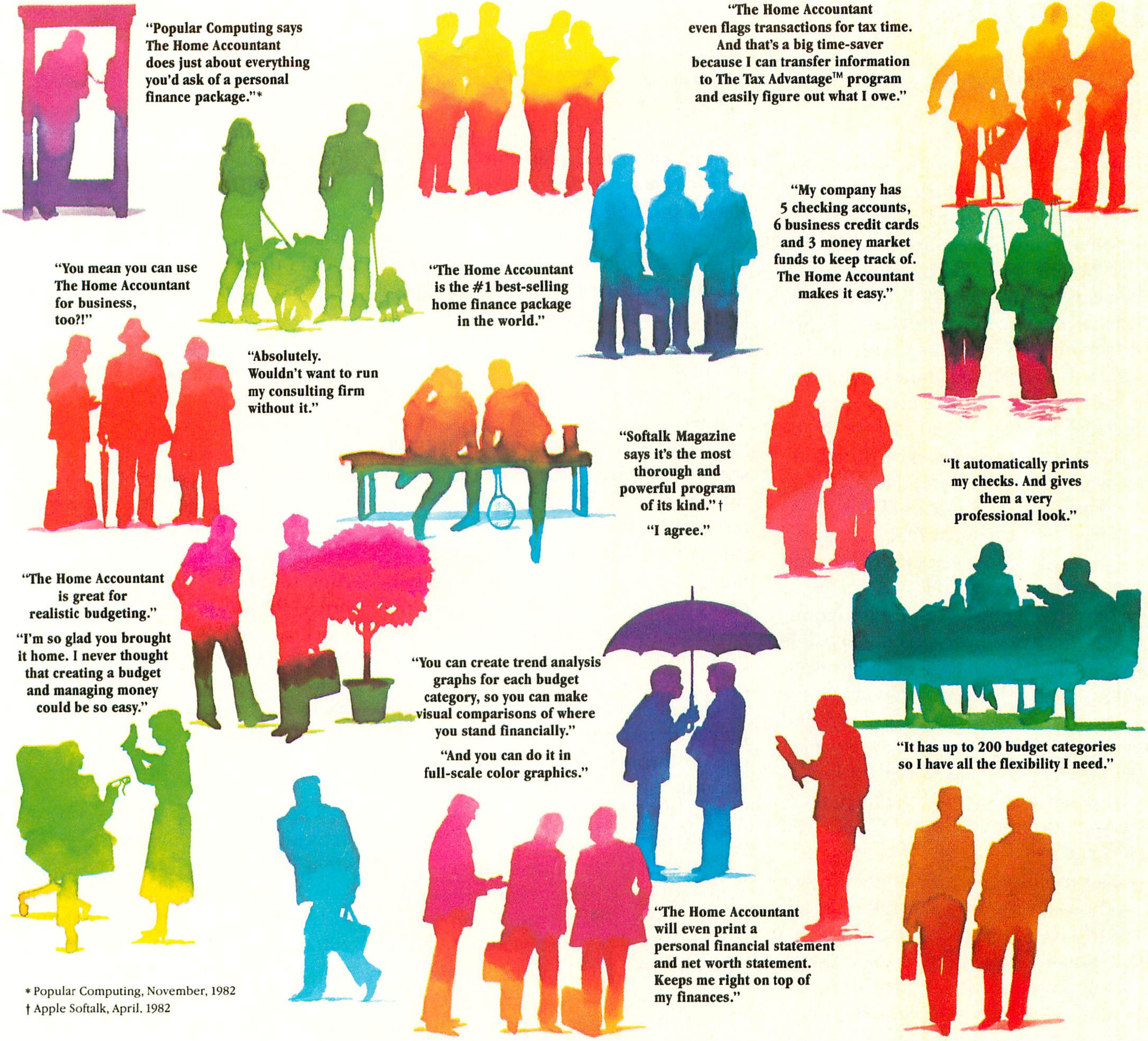
POKEY is the digital I/O (Input/Output) chip. It has a multitude of functions. The chip has four independent sound channels. Each channel has a frequency register and a control register that regulate sound volume and distortion. You access these through the SOUND command from BASIC. You can also POKE directly into these registers if you wish. POKEY reads values from the joysticks and paddle controllers. It also scans the keyboard for input, operates random numbers and handles serial I/O (i.e., communication with peripherals, such as a cassette recorder). "Serial" simply means that data is sent one bit at a time. "Parallel" means that more than one bit is sent at a time — usually eight bits in the Atari.

PIA (locations 54016–54271)

The Peripheral Interface Adapter is a specialized microprocessor chip used to control the four controller jacks (the plugs used to connect joysticks and paddles).

It is now time to explore the subterranean caverns of the Operating System itself. Let's take our flashlights and maps and explore each of the main elements of the OS.

continued on page 18



* Popular Computing, November, 1982
† Apple Softalk, April, 1982

Everyone's talking about The Home Accountant.™

Is it because it's the #1 bestselling home finance package in the world? Or because it's extremely thorough and powerful and easy to use? Or because it's great for home and business use? Or because it has up to 200 budget categories and handles up to 5 checking accounts?

Yes. But there are a lot more reasons why people buy The Home Accountant. And why you will, too.

Because The Home Accountant can literally save you hours of time. And take the headache out of handling your finances. Whether it's setting up a budget, cataloging your expenses, balancing your checkbooks or handling your credit cards and money market funds. For personal or business use.

The Home Accountant will even print net worth and financial statements. Not to mention being a lifesaver at tax time. Especially when you're able to transfer information onto Continental's The Tax Advantage™ program and figure out what you owe. Quickly.

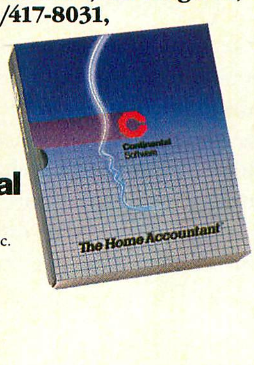
In short, The Home Accountant is the most effective software program there is for managing your money. And managing it easily.

Stop by your Continental Software dealer today and pick up The Home Accountant. You'll see what everyone's talking about.

The Home Accountant is available for Apple II/IIe, IBM PC/XT, Atari 400/800/1200XL, Osborne,® TRS-80 Models III/4, Commodore 64, Texas Instruments Professional, Zenith Z-100/110, Compaq and KayPro computers. Actual budget capacities will vary with each computer.

For your free 48 page booklet, "Tips For Buying Software," please write Continental Software, 11223 South Hindry Avenue, Los Angeles, CA 90045, 213/417-8031, 213/417-3003.

Continental Software
A Division of Arrays, Inc.



The Home Accountant and The Tax Advantage are registered trademarks of Continental Software. Apple II/IIe are registered trademarks of Apple Computer, Inc. IBM PC/XT are registered trademarks of IBM Corp. Atari 400/800/1200XL are registered trademarks of Atari, Inc., a division of Warner Communications, Inc. Osborne is a trademark of Osborne Computer Corp. TRS-80 Models III/4 are registered trademarks of Tandy, Inc. Commodore 64 is a registered trademark of Commodore Business Machines, Inc. Zenith Z-100/110 are registered trademarks of Zenith Data Systems. Texas Instruments Professional is a registered trademark of Texas Instruments, Inc. Compaq is a registered trademark of Compaq Computer Corp. KayPro is a registered trademark of Non-Linear Systems, Inc.

A CINDERELLA STORY
continued from page 16

MONITOR

Don't confuse this with a television monitor. When you turn on your computer (this is called a "coldstart"), the computer must be set up in a standard way before it can release control to a language (such as BASIC). The monitor is a machine-language program that performs this vital function. One of its first actions is to determine how much RAM it has to work with. This is done by checking the first byte of each 4K block of memory. If a value can be stored and recovered from that byte, that 4K block is assumed to be functional.

The total memory available, counted in "pages," is stored at memory location 106 (called RAMTOP). A page of memory is defined as 256 contiguous bytes whose first byte is evenly divisible by 256. To find the total memory available for your use, multiply the value found in RAMTOP by 256. You'll find that available memory is always less than the amount installed in the computer. This is because some of the memory is appropriated for use by the OS, the 6502 chip and BASIC.

The monitor performs many other functions during a coldstart. It clears all of RAM (by setting each location to zero) and initializes "pointers" (locations that contain the address of a particular table or routine). It then proceeds to initialize (set up) the device handlers. These are machine-language routines that control the editor, screen, keyboard, serial printer and cassette recorder. The monitor also determines whether a cartridge is installed and, if so, prepares for its subsequent use. Finally, it checks to see if a machine-language program needs to be "booted" in from cassette or disk.

BOOTING

The term "boot" is a piece of jargon that derives from the word "bootstrap," which was made famous by the phrase, "lifting yourself up by your own bootstraps." Bootstrapping is a technique by means of which a program brings itself into an operating state by its own action. Thus, the monitor contains a bootstrap loader, a routine that is capable of reading the initial instructions of another machine-language program. The program being read is set up so that its initial instructions cause the rest of the program to be loaded (i.e., it is a "bootable" program).

When all of this has been accomplished, the monitor releases control to the installed cartridge or the booted program. If neither is present, the Operating System goes into "memo pad" mode. In this mode, anything you type will appear on the screen, but nothing else will happen.

The monitor also handles the RESET sequence, which occurs when you press the [SYSTEM RESET] key. This is termed a "warmstart." Many of the functions of a coldstart are also performed during RESET. One of the big differences is that RESET clears only the RAM area used by the OS. Your BASIC program is *not* wiped out, as it would be if you turned off the computer.

There are a number of tricks we can play with the monitor. For example, we can go through the coldstart sequence without shutting off the computer. Simply POKE a one into location 580 and then press the [RESET] key (but remember that you'll erase any program in memory!). This has a number of uses. If you POKE this location from within a program, you can prevent users from gaining control of the program while it's running. If they press [RESET], the program will be erased and they'll have to start over. A more sophisticated use of this technique is to use it with a disk-booted program. In this case, if the [RESET] key is pressed by accident (or design), all is not lost. The coldstart sequence will automatically reboot the program.

INTERRUPT PROCESSING

The interrupt system is one of the most powerful tools available to the creative programmer. This capability is built right into the 6502 microprocessor chip.

As computer jargon goes, the term "interrupt" is descriptive. There are two special pins (electrical connections) on the 6502. When the normal voltage to either of these pins is lowered, this "interrupt" signal tells the microprocessor that some outside event is demanding its attention. The 6502 completes the instruction it is working on and then saves necessary information so that it can eventually return to the correct place within the main program. It then goes to a pair of specific memory locations where it finds the address of a machine-language routine to execute. The chip begins executing the machine-language instructions at the new address until it encounters the code for an RTI (Return from Interrupt) instruction. When the RTI is found, the chip goes back to the main program, retrieves the saved data, and then nonchalantly continues on as if nothing had happened. Since the interrupt routine is short and the microprocessor is *very* fast, the observer isn't even aware that the chip is playing hookey from the main program.

Corresponding to the two 6502 pins, there are two types of interrupts:

NMI (NonMaskable Interrupt)
IRQ (Interrupt Request)

The IRQ's can be turned off so that the chip won't respond to them. This is done by a machine-language command which, unfortunately, disables all the IRQ's. NMI interrupts cannot be turned off at the processor chip. But to allow for greater flexibility in interrupt usage, the folks at Atari routed many of the interrupts through the helper chips. For example, some NMI's can be controlled by POKE commands to certain control registers before they reach the 6502. ANTIC controls the NMI's; POKEY and PIA handle the various IRQ's.

VBI'S AND DLI'S

Two NMI's provide the bulk of the graphics and animation power that is available to machine-language programmers. These are the Vertical Blank Interrupt (VBI) and Display List Interrupt (DLI). An understanding of VBI's requires a short disgression into the workings of your television.

The television screen is lit by an electron beam that sweeps across the face of the TV tube and hits phosphors. A full

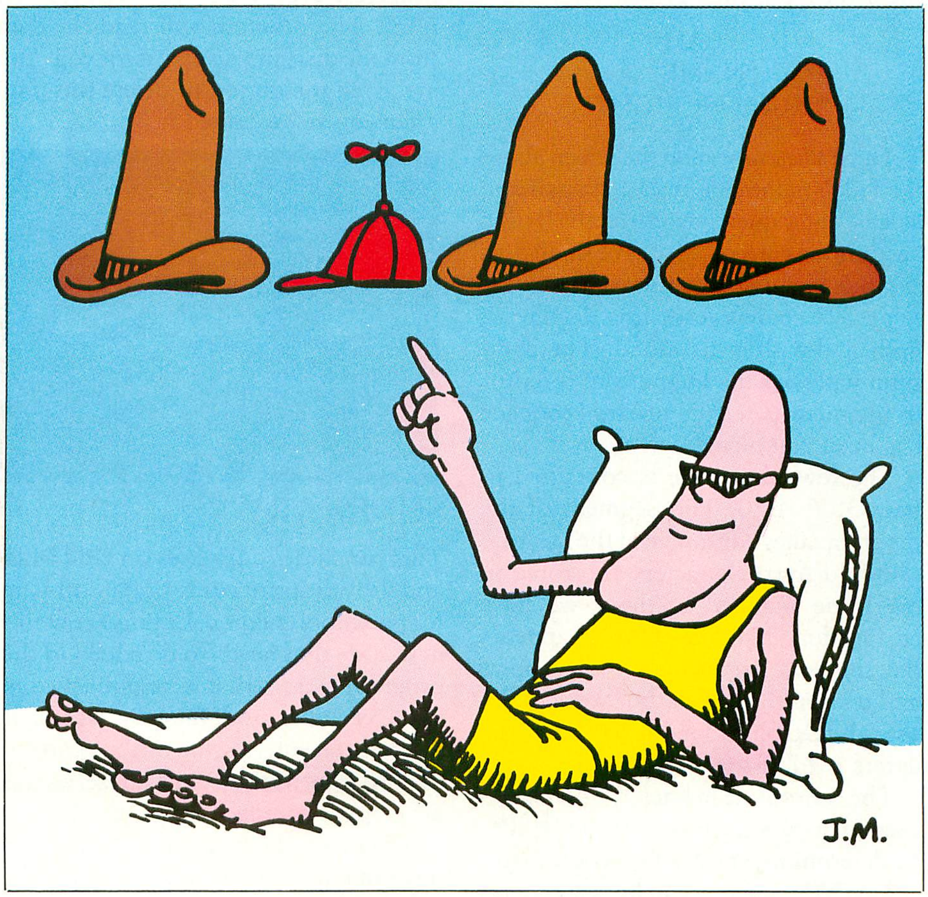
continued on page 92

THE END OF ODD MAN

Scoring and reward routines wrap it up

by JOHN and MARY HARRISON

"The End of Odd Man" is the fourth installment in a series designed to teach the basics of writing educational software. The basis of the series is a program called Odd Man Out. Portions of the program appeared in the December 1983 and January 1984 issues of ANTIC. The final portion of the program, to be merged with the previous parts, is contained in Listing 1 at the end of this article. Listing 1, therefore, is not a complete program, and will not run on its own. The RAM requirements of the completed program are 24K (cassette) or 32K (disk). —ANTIC ED



In the "Odd Man" series, we've discussed a number of techniques that can be used to improve the quality of user-written educational software. This month we'll conclude our discussion of this topic, and the program **Odd Man Out**, by covering the remainder of the program and recapping some of the highlights of the series, which began in the November 1983 issue.

THE MAKING OF ODD MAN

So far we've presented, in bits and pieces, a program designed to develop visual discrimination skills in preschool children. If we add scoring and reward routines to the portions of the program that have already appeared, Odd Man Out will be complete. Let's take a look at how this can be done.

Last month, we introduced Odd Man's joystick-movement and answer-select routine (lines 2500–2840). If we now add lines 2860–3080 (see Listing 1), the program will be able to determine if the correct shape has been selected, and to display a "happy" or "sad" face in response. In addition, the program will now keep track of the number of incorrect answers; this number is used in the reward routine.

Because the happy and sad faces are redefined characters, some additional work is required to enter the new character values. This is accomplished in lines 160, 600, 1360 and 4500 – 4600 of Listing 1. We've used the same techniques developed in "Odd Man Reforms" (ANTIC, December, 1983) to draw,

continued on next page

John and Mary Harrison are parents, teachers and Atari hobbyists. Mary teaches math and computer science at the high school level. John holds an M.S. in computer science and develops educational software. They coordinate the Education Department for ANTIC.

define and store the values for these new characters.

REWARDING A JOB WELL DONE

If a child is going to play Odd Man Out more than once, the game must give the child a sense of accomplishment. Parental praise is the child's most important reward, of course, but it never hurts to have the Atari computer offer a few kind words. Because of this, we designed a simple reward routine that incorporates the child's name. The computer clears the screen and then prints:

YEAH
"NAME"
"NUMBER RIGHT"

Three stars appear on the screen above the child's name if he or she has answered at least eight questions correctly. If nine answers are correct, a series of small dots explode in a display of Atari fireworks. A perfect score causes this display to flash and a siren to sound. The child simply presses the fire button to return to the menu after the reward sequence has been completed.

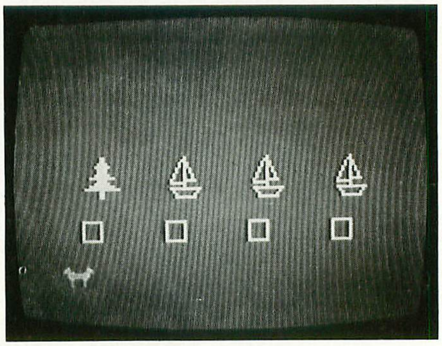
The reward routine is contained in lines 3100–4120. This segment of the program takes advantage of the fact that standard Atari characters are stored in the same manner as those we have created for Odd Man Out. It appears that the reward sequence mixes graphics and text on the screen, but we actually used the Graphics 5 mode to "draw" the letters used in the display.

The subroutine in lines 3760–3920 is used to display the text stored in DISP$. By determining the X and Y coordinates (PXO,PYO), the text can be centered on the screen. Next, each letter in the reward message is broken down into a series of numbers that describe the bit map of the character. This allows it to be plotted on the screen. The characters are originally drawn in one color. Different color registers are then used to generate the reward display's pulsating effect.

TRACKING A CHILD'S PROGRESS

One of our goals for Odd Man Out was to make it possible to monitor a child's progress in playing the game. As a result, we've provided a simple, yet effective, monitoring system as part of the program. Lines 700–760 open RECORD. DAT, the file that is used to store data concerning a child's progress. Lines 4060–4080 write this information to disk after each complete game. This occurs without the child's knowledge, and is thus not a distraction as he or she plays (and learns).

The monitoring program is contained in Listing 2. To review your child's progress, type RUN "D:MONITOR. BAS". The program will read the data from the disk and display it for you. You are given the option to delete this data after you've examined it.



SUMMARY

This concludes our series on Odd Man and the writing of educational software for preschool children. Listing 1 contains new code that needs to be added to the program listings that were published in the December and January issues. If you've been entering the various parts of the program as the Odd Man series has developed, you need only enter Listing 1 to complete the program. (The January installment explains how to merge the December and January segments of the listing.)

Now let's look back on what we set out to accomplish with the Odd Man series and how these goals were met.

In "Odd Man Out" (ANTIC, November 1983), we discussed the process of selecting an educational topic that is appropriate for the Atari computer to handle. Not every educational topic can, or should, be presented in this way; you shouldn't try to do the impossible. You should be imaginative, however.

Once you've selected a topic, you need to set specific objectives and stick to them. Programs of limited and well-defined scope are the easiest ones to write and use.

Your first step is to decide what kind of educational software you want to develop. Do you want to create a drill-and-practice program, a tutorial, a simulation, or a combination of these choices? Then, when you've finished the program, write a detailed description of what it does, and how it does it. Otherwise, you'll find yourself looking at the code six months later and wondering how and why the program was written.

In "Odd Man Reforms" (ANTIC, December 1983), we described a specific technique that we have found to be useful in the writing of educational software. The Atari computer has the ability to easily redefine character sets. This reduces the burden that is usually encountered by programmers who want to display figures on the screen. Since we used this technique extensively in Odd Man Out, we put a great deal of thought into the assignment of locations for redefined characters within the Atari character set. We wanted to ensure that the standard numerals and letters would be available as required and that they would not require any additional manipulations.

Last month, in "Odd Man At Play," we discussed simple character animation. We used this technique to add some game elements to the program to increase its level of interest for young children. In this application, the flexibility of the Atari character set was again evident.

This month's installment ties up the loose ends and concludes the listings for Odd Man Out. We hope that the techniques we've discussed will be useful to you in your own endeavors. Perhaps they'll even provide you with a starting point for that educational program you've been planning to write, but didn't know how to approach. Finally, we hope that those of you with young children will find that Odd Man Out is a worthwhile and challenging educational program, and that it will help your children develop their visual discrimination skills.

Requires 24K RAM (cassette) or 32K RAM (disk)

```
10 REM *****************************
20 REM *                           *
30 REM *        ODD MAN OUT        *
40 REM *             BY            *
50 REM *   JOHN AND MARY HARRISON  *
60 REM *                           *
70 REM *            FOR            *
80 REM *      ANTIC MAGAZINE       *
90 REM *                           *
95 REM *****************************
160 DIM FACE$(48)
600 FOR I=1 TO 48:READ X:FACE$(I,I)=CH
R$(X):NEXT I
700 TRAP 720:OPEN #4,9,0,"D:RECORD.DAT
":GOTO 760
720 REM TRAP ROUTINE IF RECORD.DAT DOE
S NOT EXIST
740 CLOSE #4:OPEN #4,8,0,"D:RECORD.DAT
"
760 TRAP 40000
780 REM INPUT CHILD'S NAME
800 GRAPHICS 0:OPEN #5,4,0,"E:"
820 POSITION 2,5:? "Hello.  What is yo
ur name? ";:INPUT #5;NAME$
840 POSITION 2,7:? "Plug your joystick
 into port number 1 and press the fire
 button to begin."
860 IF STRIG(0)<>0 THEN 860
1070 FOR DEL=1 TO 25:NEXT DEL
1360 RAM$(209,256)=FACE$(1,48)
2860 POSITION INT(LE)*4+3+(CHOICE=7),8
:? #6;"\"
2880 POSITION 9,1:? #6;":;"
2900 IF INT(LE)<>ODD THEN 3000
2920 POSITION 9,2:? #6;"<="
2940 FOR K=121 TO 0 STEP -5:SOUND 0,K,
12,8:SOUND 1,K+5,12,8:NEXT K:SOUND 0,0
,0,0:SOUND 1,0,0,0
2960 FOR DEL=1 TO 75:NEXT DEL
2980 POSITION 9,1:? #6;"  ":POSITION 9
,2:? #6;"  ":? "⇱":RETURN
3000 POSITION 9,2:? #6;">?"
3020 FOR K=1 TO 30:SOUND 0,60,12,7:SOU
ND 0,60,12,8:NEXT K:SOUND 0,0,0,0
3040 IF FLAG=0 THEN WRONG=WRONG+1:FLAG
=1
3060 POSITION 9,1:? #6;"  ":POSITION 9
,2:? #6;"  "
3080 POSITION INT(LE)*4+3+(CHOICE=7),8
:? #6;"[":? "⇱":GOTO 2540
3100 REM REWARD ROUTINE
3120 POKE AF,0
3140 GRAPHICS 21:FOR I=0 TO 2:SETCOLOR
 I,4,6:NEXT I:SETCOLOR 4,7,0:C=1
3160 DISP$="YEAH":PYO=22:GOSUB 3760
3180 DISP$=NAME$:PYO=31:GOSUB 3760
3200 IF WRONG=0 THEN 3240
3220 DISP$="  RIGHT":DISP$(1,1)=STR$(1
0-WRONG):GOTO 3260
3240 DISP$="PERFECT"
3260 PYO=40:GOSUB 3760
3280 REM ROUTINE FOR ATLEAST 8 RIGHT
3300 IF WRONG>2 THEN 4060
3320 COLOR 1:PLOT 40,0:DRAWTO 40,10:PL
OT 35,5:DRAWTO 45,5
3340 COLOR 2:PLOT 42,3:DRAWTO 38,7:PLO
T 38,3:DRAWTO 42,7
3360 REM ROUTINE FOR ATLEAST 9 RIGHT
3380 IF WRONG>1 THEN 4060
3400 COLOR 1:PLOT 10,12:DRAWTO 20,12:P
LOT 15,7:DRAWTO 15,17
3420 COLOR 2:PLOT 17,9:DRAWTO 13,15:PL
OT 13,9:DRAWTO 17,15
3440 COLOR 1:PLOT 60,12:DRAWTO 70,12:P
LOT 65,7:DRAWTO 65,17
3460 COLOR 2:PLOT 67,9:DRAWTO 63,15:PL
OT 63,9:DRAWTO 67,15
3480 REM ROUTINE FOR 10 RIGHT
3500 IF WRONG>0 THEN 4060
3520 SETCOLOR 2,3,4:COLOR 3
3540 FOR LP=1 TO 15:X=INT(80*RND(0)):Y
=INT(20*RND(0))
3560 GOSUB 3940:PLOT X,Y:NEXT LP
3580 REM SOUND EFFECTS FOR PERFECT SCO
RE
3600 LO=50
3620 FOR I=1 TO 20:FOR J=0 TO 13
3640 SETCOLOR 0,J,8
3660 SETCOLOR 1,J+1,8
3680 SETCOLOR 2,J+2,8
3700 SOUND 0,LO,10,14:LO=LO-1:IF LO<35
 THEN LO=50
3720 NEXT J:NEXT I:SOUND 0,0,0,0
3740 GOTO 4060
3760 PXO=(80-LEN(DISP$)*10)/2:PX=PXO:C
=1:COLOR C
3780 FOR I=1 TO LEN(DISP$):PY=PYO
3800 CHAR=ASC(DISP$(I,I))-32:PTR=CHAR*
8+1
3820 FOR J=PTR TO PTR+7:BITM=ASC(ROM$(
J,J)):DIV=128
3840 FOR K=1 TO 8
3860 IF BITM/DIV>=1 THEN PLOT PX,PY:BI
TM=BITM-DIV
3880 DIV=DIV/2:PX=PX+1:NEXT K:PX=PXO:P
Y=PY+1
3900 C=3-C
3920 COLOR C:NEXT J:PXO=PXO+10:PY=14:C
OLOR 1:NEXT I:RETURN
3940 REM EXPLOSION SOUND
3960 SOUND 2,75,8,15:V1=15:V2=15:V3=15
:ICR=0.079
3980 SOUND 0,20,8,V1:SOUND 1,40,8,V2:S
OUND 2,70,8,V3
4000 V1=V1*ICR:V2=V2*(ICR+0.05):V3=V3*
(ICR+0.08)
4020 IF V3>1 THEN 3980
4040 SOUND 0,0,0,0:SOUND 1,0,0,0:SOUND
 2,0,0,0:RETURN
```

continued on next page

```
4060 REM CONDUCT RECORD KEEPING AND LO
OK FOR JOYSTICK INPUT TO CONTINUE
4080 ? #4;NAME$:? #4;CHOICE:? #4;10-WR
ONG
4100 IF STRIG(0)<>0 THEN 4100
4120 GOTO 880
4140 GRAPHICS 0:CLOSE #4:END
4160 REM
4500 DATA 0,7,24,32,76,76,64,64
4520 DATA 0,224,24,4,50,50,2,2
4540 DATA 64,72,68,67,32,24,7,0
4560 DATA 2,10,18,226,4,24,224,0
4580 DATA 64,67,68,70,32,24,7,0
4600 DATA 2,194,34,18,4,24,224,0
```

TYPO TABLE

Variable checksum = 859076

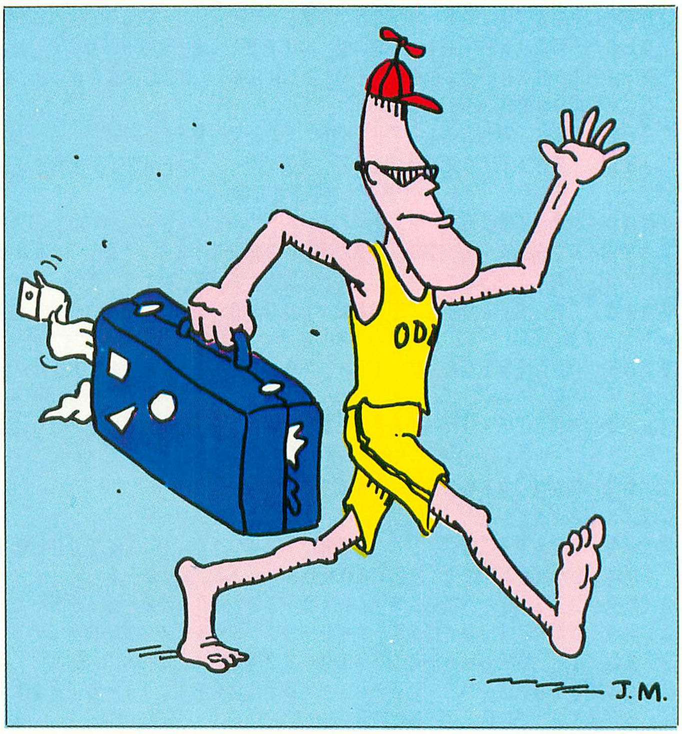
Line num range	Code	Length
10 - 600	HL	401
700 - 1360	VM	518
2860 - 3020	QI	596
3040 - 3260	LX	512
3280 - 3440	VQ	537
3460 - 3680	VC	451
3700 - 3920	GI	485
3940 - 4100	RM	521
4120 - 4600	LU	209

```
10 REM ******************************
20 REM *                            *
30 REM *    ODD MAN OUT MONITOR     *
40 REM *                            *
50 REM *            BY              *
60 REM *                            *
70 REM *  JOHN AND MARY HARRISON    *
80 REM *                            *
90 REM *           FOR              *
100 REM *                            *
110 REM *       ANTIC MAGAZINE       *
120 REM *                            *
130 REM ******************************
140 DIM NAME$(20),Z$(5)
150 OPEN #5,4,0,"E:"
160 TRAP 320
170 OPEN #2,4,0,"D:RECORD.DAT"
180 TRAP 40000
190 GRAPHICS 0:POKE 752,1:REM POKE 16,
112:POKE 53774,112
200 POSITION 11,1:? "ODD MAN OUT REPOR
T"
210 POSITION 2,3:? "NAME":POSITION 20,
3:? "LEVEL       SCORE":?
220 TRAP 350
230 CTR=0
240 INPUT #2;NAME$:INPUT #2;LEVEL:INPU
T #2;SCORE
250 X=20-LEN(NAME$)
260 ? NAME$;:POKE 201,X:? ,LEVEL;:POKE
 201,9:? ,SCORE
270 CTR=CTR+1
280 IF CTR<15 THEN 240
290 ? "Press RETURN to contiue.";
300 INPUT #5;Z$
310 ? "↖":GOTO 200
320 REM FILE DOES NOT EXIST TRAP
330 IF PEEK(195)<>170 THEN ? "ERROR -
";PEEK(195);" IN LINE 170":CLOSE #2:EN
D
340 ? "FILE RECORD.DAT DOES NOT EXIST.
  UNABLE TO PROVIDE REPORT.":CLOSE #2:
END
350 REM OUT OF DATA TRAP
360 POKE 84,20:POKE 201,10:POKE 752,0
370 IF PEEK(195)<>136 THEN ? "ERROR -
";PEEK(195);" IN LINE ";PEEK(186)+256*
PEEK(187)
380 CLOSE #2
390 ? "Press RETURN to continue"
400 INPUT #5;Z$:? "↖"
410 ? "Do you want to delete the store
d      information? ";:INPUT #5;Z$
420 IF Z$(1,1)="Y" THEN OPEN #2,8,0,"D
:RECORD.DAT":CLOSE #2
430 END
```

TYPO TABLE

Variable checksum = 147064

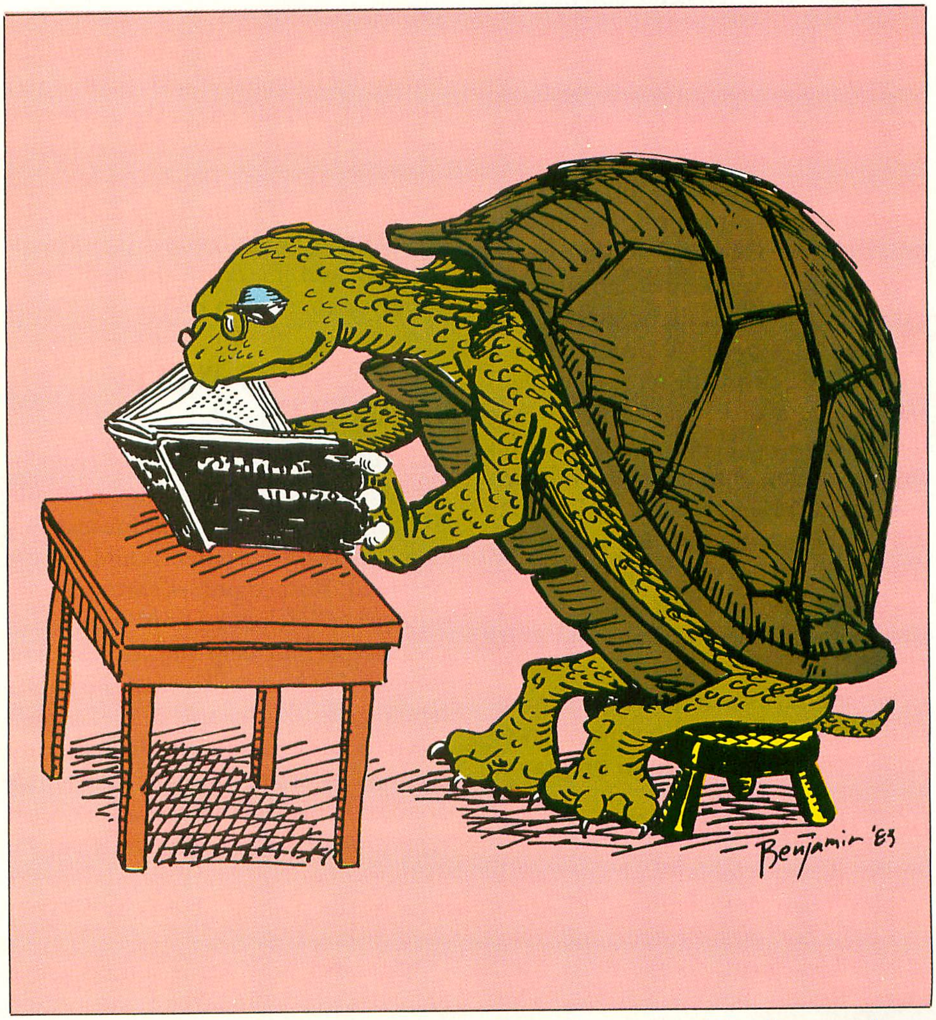
Line num range	Code	Length
10 - 120	VB	420
130 - 240	BM	413
250 - 360	QW	443
370 - 430	BL	318



LOGO BOOKS

The lowdown on Atari Logo

by KEN HARMS



When Atari decided to implement Logo, it chose to make its version compatible with Apple Logo, which is produced by LCSI and sold by Apple. In fact, LCSI also wrote Atari's Logo. Although this means that we're forced to live with a few Apple-like quirks in our version of Logo, it also means that virtually all of Apple's Logo programs run under Atari Logo. In addition, most of the LCSI-oriented Logo books also fit our needs. Just as Atari intended, this element of compatibility between the two languages provides Atari users with access to a wealth of classroom-support materials, books, and other sources of valuable information.

ATARI LOGO'S KINFOLK

This month, I'll delve into a stack of Logo books and highlight the ones you may want to consider for your classroom or family. But first, a warning. Although all Logos are related, some relatives are closer than others.

Atari Logo's closest relative is Apple Logo. Books based on this language (often referred to as "LCSI-Logo") are 99% "upward compatible." In effect, this means that Atari Logo can do everything that Apple Logo can do. But Atari Logo also offers extra features that Apple Logo doesn't — including animated turtles, multiple turtles, and more.

The Logo books produced by Krell/Terrapin are Atari Logo's next closest relation. TI's Logo requires even more translation. And the TRS Color Computer's Logo isn't even a full Logo; it's

continued on next page

Ken Harms, our Contributing Editor for the Logo/PILOT department, is Vice President of Administration for the California Division of the American Cancer Society.

just a turtle-graphics package. My judgment is that the average Atari user will find that the LCSI/Apple Logo books are completely satisfactory, while the Krell/Terrapin books are acceptable. Personally, I'd leave the rest of the books alone, except those that have sections which explain how to translate material into either Apple or Atari Logo.

A LITTLE HELP FROM YOUR FRIENDS

One of the best support sources for Atari Logo users is the Young People's Logo Association (1208 Hillsdale Dr., Richardson, Texas 75081). A membership ($9/year for students 18 or younger, $25/year for adults) provides you with a hefty monthly newsleter, software exchange privileges and access to a bulletin board.

All of these materials are dedicated exclusively to Logo/PILOT. Although it supports several machines, each issue of the newsletter is loaded with Atari Logo programs and information, and the software exchange includes 30 disks of Atari Logo programs. A member can purchase a disk (or tape) for $10, or, even better, can send in a personal program on a disk/tape (along with return postage) and receive a disk/tape of his or her choice with 12 to 15 programs on it. And anyone can log onto the bulletin board by calling (214) 783-7548 after 7:00 pm (CST). There's no cost for use of the board (but you pay your own phone bills, of course). It lists a number of special interest groups for Atari Logo and Atari PILOT. The board is currently set up to accept messages and handle electronic mail, but it will soon allow members to upload and download Atari programs.

BOOKS FROM YPLA

Jim Muller, the founder of the Young People's Logo Association (YPLA), is also coauthor, with Bearden and Martin, of *The Turtle Source Book* (200 pages, Reston Books, $21.95). This excellent workbook-style manual is designed for the elementary classroom but can probably be used by most families, if an interested adult is present. The books does a good job of covering turtle graphics, and is one of the few books on the subject whose language level is appropriate for youngsters.

LIST PROCESSING AND MORE

Apple Logo by Harold Abelson (217 pages, Byte/Neforan Hill, $15.95) is one of the few books on Logo that covers list processing as well as turtle graphics. Although Atari users will have to convert the sections on I/O and paddle use to Atari Logo, Abelson provides us with a rich vein of new words (commands) for Logo, and a lucid explanation of advanced list handling. He also covers numerical calculations and program-logic controls by means of examples that I found helpful.

All in all, this is a well-rounded, solid book; the language level is high school or above, and the pages are dense with text. I'm hoping to see an Atari version soon. (By the way, watch out for *Logo for the Apple II* by the same author. It's based on MIT's Logo and is not quite as useful as *Apple Logo*, although it does supply a conversion table).

LEARNING WITH ATARI LOGO

Learning with (Atari) Apple Logo by Daniel Watt (358 pages, McGraw Hill, $19.95) is an excellent text for teaching Logo to a class or a parent, child, spouse or friend. Watt combines cartoons with numerous "Helper's Hints" that teach the "helper" to help the learner. The result is that both will learn. The book's language level — simple in the cartoon sections and adult in the Helper's Hint sections — is appropriate. Like *Apple Logo*, it helps you to build an excellent repertoire of words (commands), and it includes one of the best explanations I've seen of the names, objects and values used in Logo.

Watt also provides a simple game, SHOOT, which should be of interest and value to the budding Logo programmer.

This book is well illustrated and contains many interesting projects and pieces of good advice for teachers. I wasn't able to find the Atari version on the book shelves, but I was impressed with the Apple text. Watt's *Learning with Logo* is based on Krell's Logo and, therefore, isn't as helpful, but it's still useful if it's all you can find.

The *Source Book* also includes off-computer exercises that help to introduce the concepts of turtle graphics. In addition, a number of stickers are included with the workbook; for example, it includes numerals that can be attached to the CRT screen to make a clock/time-reading program come alive. Finally, where appropriate, the book's activity pages are reproducible, so that a full classroom can use a single source book.

Although I haven't seen a copy yet, Reston Books should have a second YPLA book out by the time this column reaches you. *123 My Computer and Me* (Reston Books, $10.95), by Donna Bearden, is a "fun" companion to the *Source Book*. It's designed for the individual Logo explorer in the eight-to-twelve age group. Like the *Source Book*, it stresses turtle graphics.

DISCOVERING ATARI LOGO

Discovering Apple (Atari) Logo by David D. Thornburg (173 pages, Addison-Wesley, $14.95) is probably the most complete, *and* the most beautiful, treatment of turtle graphics you can find. An Atari version of the book should be available by the time you read this, but it wasn't in yet at any of the book stores I visited.

Thornburg explores the Logo user's world of shapes and planes with an eye that is always sensitive to the "larger meaning of things." Although some passages may be difficult for youngsters, all will benefit from the author's insights and will enjoy the beautiful patterns that his always-simple programs create. Thronburg covers many advanced topics, including "tiles," tessellations, "squrals," fractals, trees, arcs, cricles, and spirals, and he makes them both understandable and inspiring. He's also the only person I know of who's come up with a convincing explanation of why you need to know about prime numbers: you can use them to draw stars! Twelve

color photos enhance an already enjoyable book.

A LOGO PRIMER

Apple Logo Primer by Gary Bitter and Nancy Watson (206 pages, Reston Books, $14.95) offers a unique approach: It includes a section for the complete novice, one for people who are generally familiar with Logo, and a third section on the history of the language. Graphic outputs are illustrated by screen photos (most books use printed graphics) that are realistic but often muddy. The section on list processing is short but helpful.

One of the book's nice touches is that the authors assign memorable words, such as "Open a blank" for Control O, to help users learn Logo's "weird" control commands. The practice activities made me feel like I was back in school. The language is occasionally tough for kids — "alternative" is used instead of "choice," and "identical results" is used in place of "the same thing" — but advanced sixth graders should be able to handle it. The primer's strongest feature, in my opinion, is an excellent index/glossary that often tells you what you need to know without forcing you to turn to the referenced pages. Its excellent cover-reference cards round out a thoughtful approach.

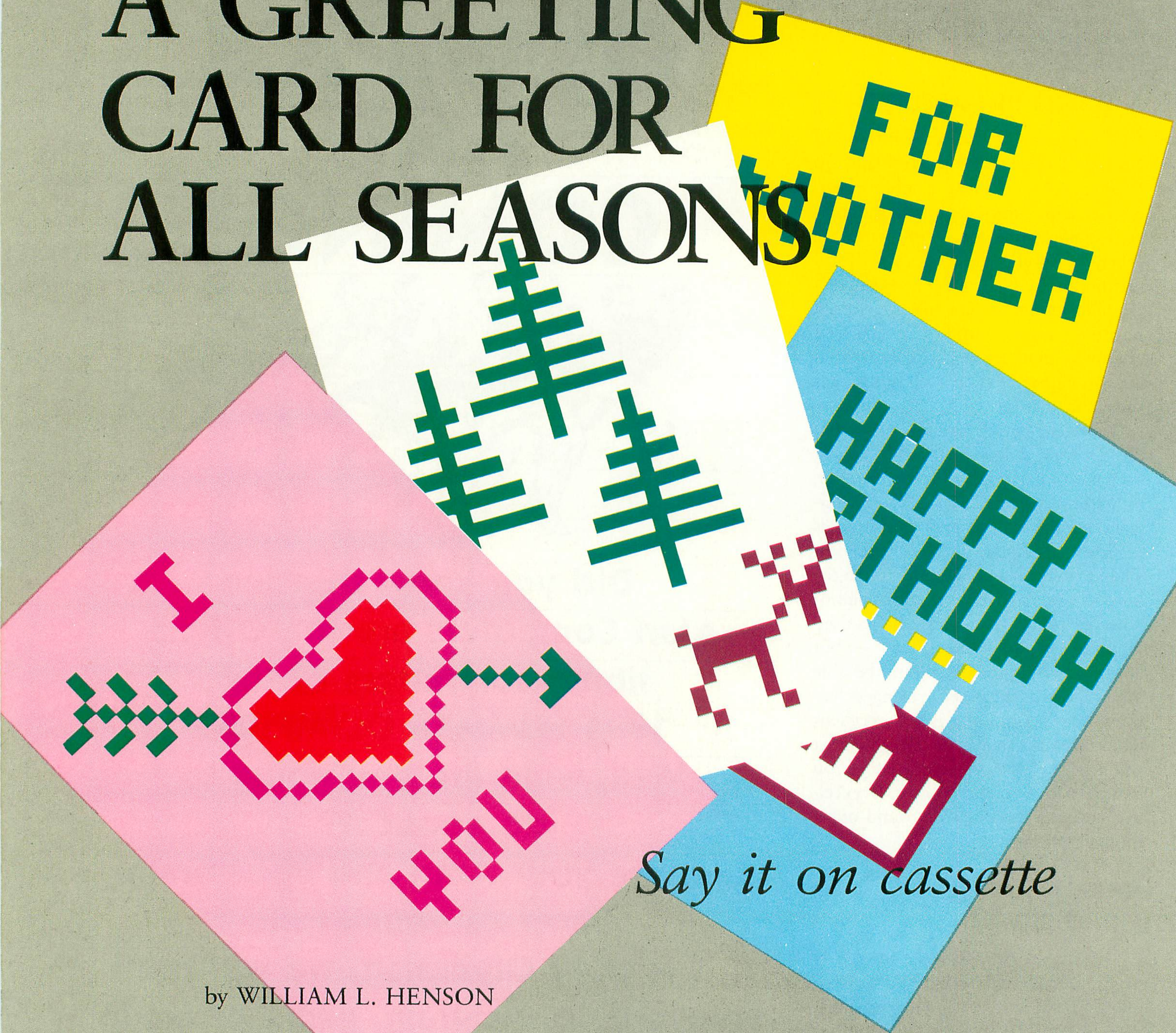
NON-LOVE AT FIRST SIGHT

Introductory Logo by Peter Ross (241 pages, Addison-Wesley, $12.95) looked interesting at first sight. Unfortunately, that's as far as I got. Its dot-matrix printing looked so ugly that I didn't read the text.

SUMMARY

Several excellent Logo texts are currently available, and more are on the way. In particular, you should watch for Atari-specific editions of existing Apple books. I'll continue to review materials as they become available. In the meantime, the books I've mentioned this month should provide a good foundation for your Atari Logo library.

A GREETING CARD FOR ALL SEASONS



Say it on cassette

by WILLIAM L. HENSON

Did you ever wake up in the morning with the sinking sensation that it's Valentine's Day (or Mother's Day, or a special birthday) and you've forgotten to buy a card for someone? Or worse, did you realize that the loved one in question would attribute your dereliction of duty to your having spent too many hours in front of your Atari? The only solution to this dilemma is to leap out of bed and resort to the All-Occasion Card. This life-saving program allows you to make a selection from ten different backgrounds, and permits you to write a message of up to three lines of horizontally scrolling text.

After you LOAD the program, your first task is to assign characters to A$, B$ and C$ in lines 460–480. For the best results, these line lengths should not be changed and your messages should be centered.

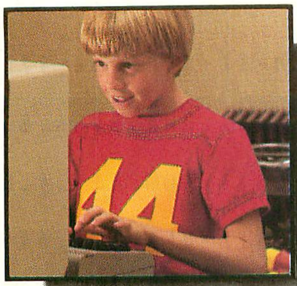
Your next task is to RUN the program. Each data line has a descriptive string leader that describes the character it creates. Lines 250–280 read and print these labels with their corresponding line numbers. You're then asked to enter the line number of the background you wish to use.

Lines 320–390 relocate the Graphics 1 and 2 character set

continued on page 28

William L. Henson is an attorney who specializes in contract disputes. He became hooked on personal computers after meeting Chris Crawford in the summer of 1978. An Atari 800 joined the family in 1982. His wife Julie and five-year-old daughter Elizabeth are also Atari enthusiasts.

Computers may be the world's best spellers, but they're the world's worst spelling teachers.



Or at least they used to be. Until Chatterbee.

Chatterbee is a unique piece of home computer software that not only makes teaching possible, it makes learning fun. For little people. And big people, too.

If that sounds like something you've never heard of before, you're right. You haven't. Because Chatterbee does something that no other spelling software can do.

Talk.

It's equipped with a customized version of the S.A.M. speech synthesizer, developed by Don't Ask for Tronix to provide voice output.

Instead of drawing pictures, scrambling words, beating around the bush or playing "Hangman," Chatterbee teaches spelling the sensible, old-fashioned way.

Each word is spoken, then put in a contextual sentence. Just like a spelling bee. With one important difference.

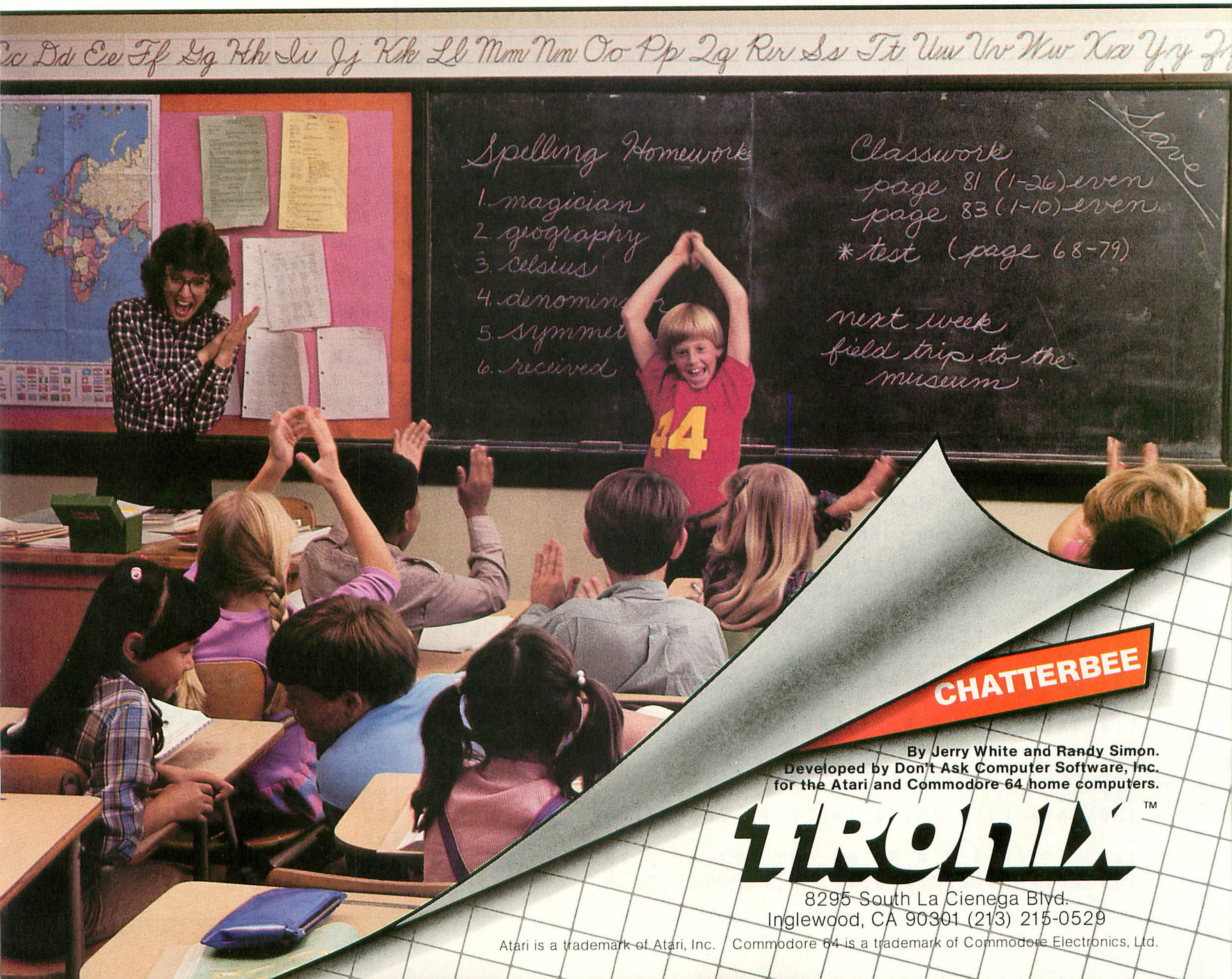
Since there's no classroom, there's no pressure. And no embarrassment.

Each of the 2500 words have been selected from current educational literature. And there's an internal scoring system that automatically places each player at the right level of difficulty and challenge.

The external scoring system rewards success and turns the whole thing into a game, complete with music.

Chatterbee. The spelling bee without the sting.

Hear today, spell tomorrow.



to RAM. Line 360 calculates the address of the background character (or space) for the lower-case graphics set. Line 410 discards the data label. Lines 420–440 customize the background character with chosen data. The last number in each data line selects a color for the character. This number is POKEd into Color Register 0.

Line 490 selects lower case. Line 500 allows the background color to vary randomly (if the REM is removed). Line 510 varies the text color. Luminances for text and background are set to ensure that the text never "merges" with the background.

Lines 520–630 print the message strings. The loop creates the illusion that messages scroll from right to left. This scrolling is a bit uneven, but the card's recipient is usually too pleased to notice this. (The elimination of one message line and variation of the delay loop at line 620 may help matters somewhat.) Three message lines seem to be the limit for simple BASIC.

Of course, there is no reason that other characters cannot be used. For purposes of experimentation, you can modify any line of data. For example, draw an 8 × 8 block on a piece of graph paper. Color the blocks in any way you wish. Treat each line as a binary number (empty = 0, full = 1). Translate each line to decimal form, and list a data line. Then substitute your new numbers for the first eight numbers in the data line.

To experiment with color, change the last number in the data line. Replace the old label with your new one. Then RUN the program and choose your line.

If you wish to use Graphics 1, subtract one from line 350. To "proportion" the page, add three to *each* POSITION statement in lines 530–610.

Any good thing can be run into the ground, of course. But by varying the backgrounds on the All-Occasion Card, you can ensure that it won't grow stale too soon. The number of times you should use the card depends on two factors: How many Atari owners you know, and how brave you are. In addition, the All-Occasion Card can be used as an interesting display at meetings and other social gatherings.

The most economical means of delivering the card is via cassette. Don't forget to provide loading instructions. And remember, on those special occasions when disaster seems to be guaranteed, the All-Occasion Card is here to help.

Requires 16K RAM

```
95 REM * ALL OCCASION CARD *
96 REM * BY WILLIAM HENSON *
97 REM *   ANTIC MAGAZINE   *
98 REM *   FEBUARY, 1984    *
99 REM **********************
100 DATA SMILE,0,60,90,126,90,102,60,0
,26
110 DATA TREE,0,0,24,60,126,24,24,0,20
0
120 DATA CAKE,0,0,42,42,42,127,127,127
,57
130 DATA LOGO,28,28,28,28,42,73,0,0,25
0
140 DATA CAR,0,0,56,68,127,127,34,0,70
150 DATA PICKUP,0,0,12,10,127,127,34,0
,87
160 DATA VAN,0,0,126,125,127,127,34,0,
156
170 DATA SHAMROCK,0,24,24,124,124,16,8
,0,212
180 DATA PUMPKIN,4,8,60,90,126,66,60,0
,38
190 DATA HEART,0,54,127,127,62,28,8,0,
52
200 DIM A$(20)
210 DIM B$(20)
220 DIM C$(20)
230 DIM D$(20)
240 GRAPHICS 0
250 FOR C=100 TO 190 STEP 10
260 RESTORE C
270 READ C$:PRINT C;"  ";C$
280 NEXT C
290 PRINT "SELECT BACKGROUND #"
300 INPUT X
310 RESTORE X
320 RAMTOP=PEEK(106)
330 CBASE=RAMTOP-4
340 POKE 106,CBASE
350 GRAPHICS 18
360 CADR=CBASE*256
370 FOR C=0 TO 1023
380 POKE CADR+C,PEEK(57344+C)
390 NEXT C
400 SPACE=CADR+64*8
410 READ C$
420 FOR C=0 TO 7
430 READ DTA:POKE (SPACE+C),DTA
440 NEXT C:READ DTA:POKE 708,DTA
450 D$=" ":D$(20)=D$:D$(2)=D$
460 A$="  happy mother's day   "
470 B$="         from          "
480 C$="  elizabeth and bill   "
490 POKE 756,CBASE+2
500 REM SETCOLOR 0,INT((14*RND(1))),2
510 SETCOLOR 1,INT((14*RND(1))),8
520 FOR X=1 TO 19
530 POSITION 0,3
540 ? #6;A$(X);:? #6;A$(1,X)
550 POSITION 0,4:? #6;D$
560 POSITION 0,6
570 ? #6;B$(X);:? #6;B$(1,X)
580 POSITION 0,7:? #6;D$
590 POSITION 0,9
600 ? #6;C$(X);:? #6;C$(1,X)
610 POSITION 0,10:? #6;D$
620 FOR L=1 TO 55:NEXT L
630 NEXT X
640 GOTO 500
```

TYPO TABLE

Variable checksum = 230327

Line num range	Code	Length
95 – 160	KB	394
170 – 280	YY	255
290 – 400	DM	198
410 – 520	LW	336
530 – 640	QY	344

THE GREATEST ATARI® GAME OF ALL TIME.

Object: Capture more programs than from any other source. Score: The best prices for programs win.

Actually, every member of CompuClub is a winner, because no one pays lower prices for Atari® programs than our members.

And no other source offers as many programs, with a descriptive catalog covering every piece of software we offer!

CompuClub has hundreds of Atari® programs: games, education and business. It's an astonishing selection, but just as incredible are our prices and our catalog.

Prices always at least 25% below retail

Fact is, usually our prices on selected programs are even lower than that. And we're always running sales with savings of at least 50% from the list price on some of your favorite programs.

And there are several ways to pay: MasterCard, Visa, or the ever popular check or money order.

Annotated catalog, updated every 45 days

We don't like to knock the competition, even by implication, but there are definite advantages to a CompuClub membership. One of the best of them is our catalog, which not only offers an unbelievable number of programs, but includes a description of <u>each and every one of them.</u>

In fact, the catalog is so big and filled with so much information that we print a separate price list. And we keep adding so many programs that we have to update the catalog nine times a year during the course of your membership.

Think of it! No more wondering what's behind the fancy label, no more shooting in the dark or depending on the word of a clerk who may not be familiar with the program you're interested in.

Exclusivity for Atari® owners

We're not trying to be all things to all people. We are definitely the best thing since the floppy disk for Atari® owners, and only Atari® owners. CompuClub is very exclusive.

Our catalog is thick with hundreds of Atari® programs, and <u>only</u> Atari® programs. No more fumbling your way through thickets of strange symbols and codes for the different computers everybody else's catalogs try to cover.

The rules

The rules are simple. To play the CompuClub "game," you've got to be a member. Anyone can join, as long as they fork over a mere five bucks, and agree to buy three programs during the year of membership. Order and buy your programs at any time during the year, but we're sure with our selection and prices that you'll want to get going right away.

Five dollars buys you a one-year membership, exclusive Atari® program offerings, a fully annotated catalog with 9 updates during the year, a current price list (and any necessary updates), a subscription to our newsletter, and discounted prices at all times, including announcements of our periodic sales, which feature savings of 50% off and more.

You can join by filling out the coupon below, or get an instant membership by calling our toll-free number and giving us your MasterCard or Visa identifying number. We'll forward your registration and sales order materials, your catalog and price list, so you can begin ordering right away.

That's all there is to it.

If you think you play a good game, you've got to join CompuClub. It's easy to play, and it saves you money.

We think it's the best Atari® game going, because with us, everybody wins.

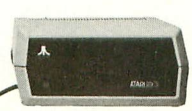
CompuClub
Where Atari owners belong.



YES, **CompuClub,** PO Box 652, Natick MA 01760 **ATTN: Dept. AT**
I want to play the greatest Atari® game of all. Please enroll me in CompuClub for one year, and rush my registration material, catalog and price list to me. I agree to buy any three programs during the course of my year of membership. I have enclosed my payment of $5, or authorization to bill my charge card account.

Please make your check payable to CompuClub
Payment enclosed ☐ check ☐ money order.
Bill my ☐ MasterCard ☐ Visa ID #__________

Expires__________



Signature__________
Name__________
Address__________
City______ State______ Zip______



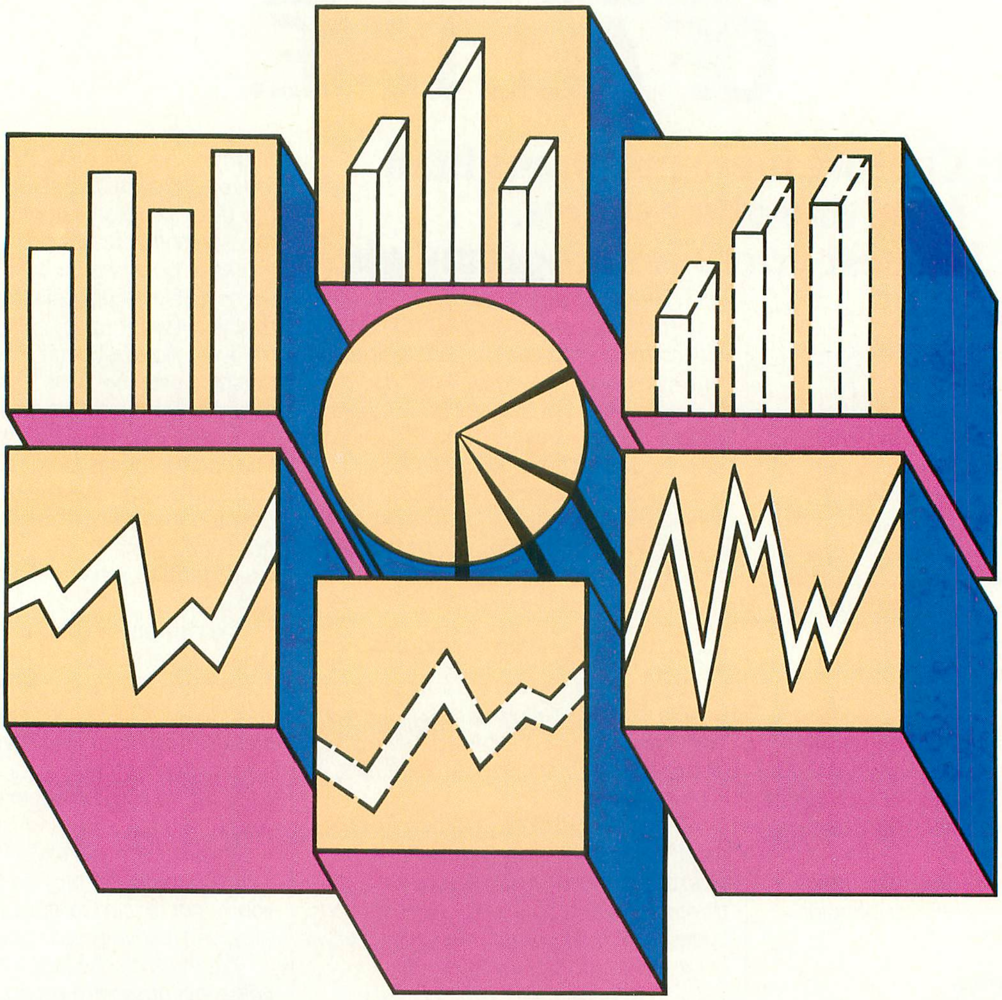
Computer Model__________ Disk ☐ Tape ☐

If you are not satisfied in any way with the conditions of your membership, you may cancel before ordering any programs from CompuClub. Your membership fee of $5 will be cheerfully refunded.

To join by phone, call toll-free.
800-631-3111

Please have your MasterCard or Visa card ready when you call. In Massachusetts, call 1-617-879-5232. We will forward your membership materials after confirming authorization for your charge.

B/GRAPH T.M.



A PROFESSIONAL GRAPHICS-CHARTING AND STATISTICAL ANALYSIS PROGRAM FOR ATARI® PERSONAL COMPUTERS

B/Graph is for professionals in Marketing Sales, Administration, Forecasting, Accounting and General Management.
- Home and Small Business
- Educators, Students
- Hobbyists

B/Graph can create
- Bar charts
- 3-D Bar Charts
- Segmented Bar charts
- Pie charts
- Line Graphs
- Scatter Graphs
- Market Graphs (High-Low-Close)

Instant conversion between all graph types without re-entering data.

B/Graph can perform
- Statistical analysis
- Regression analysis
- Exponential smoothing
- Moving averages
- Graphing of 1–3 factors with up to 100 data points.

Save graphs as data files or high-res screens.
B/Graph Features
- Automatic labelling
- Re-labelling
- Custom labelling
- Full screen editor
- Multiple graph overlays
- Automatic or manual labelling
- Automatic and manual scaling
- Automated slide shows
- Grid overlays
- Multiple graph overlays
- Total screen color control

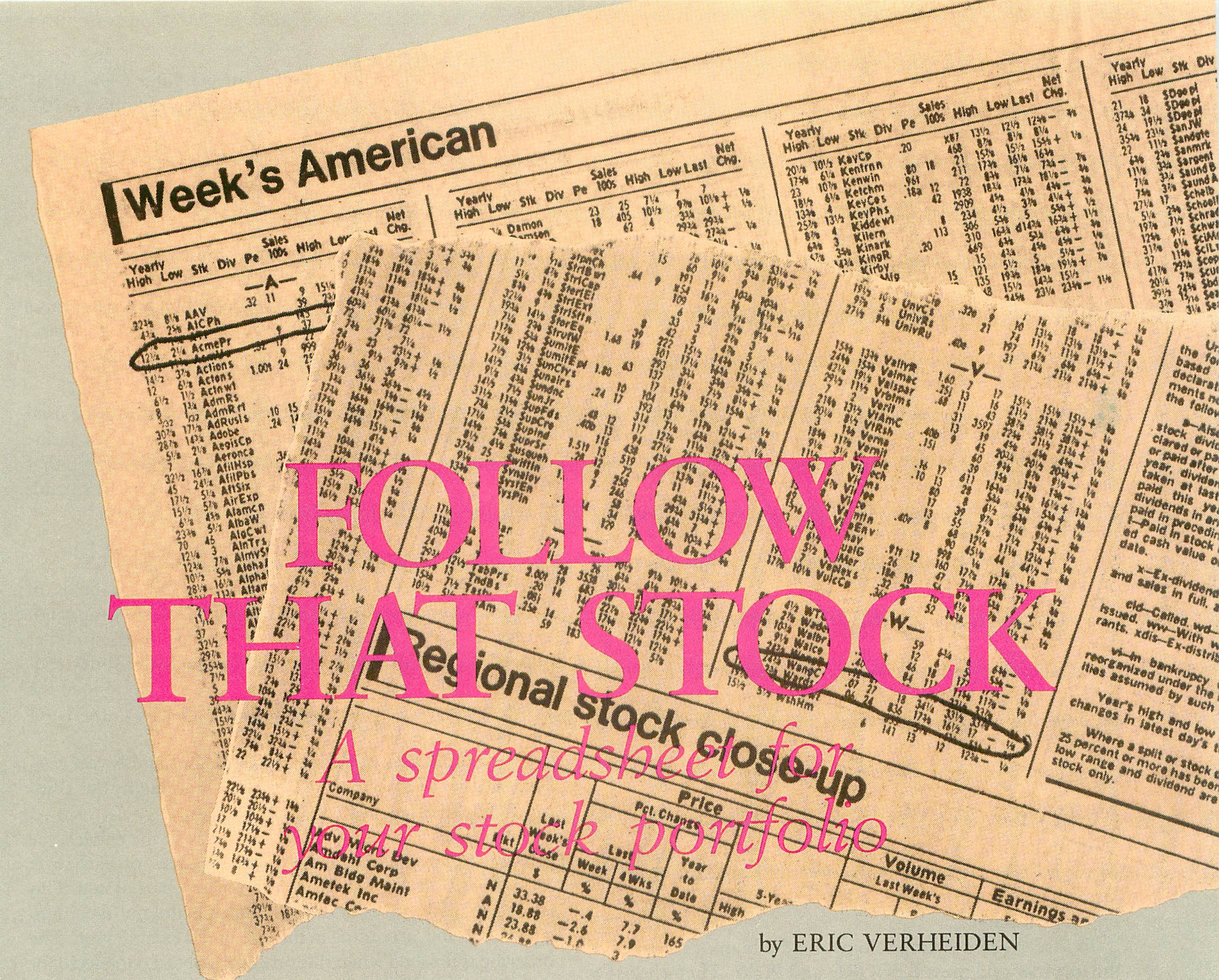
B/Graph works with graphics printers from
- Centronics
- Epson/Prowriter
- C. Itoh/Nec
- Seikosha
- Visicalc D.I.F. files
- User written programs

B/Graph comes with comprehensive professionally written documentation. A complete tutorial on Graphing/Charting and Statistics.

User registration with full program support.

INHOME

INHOME SOFTWARE INCORPORATED
2485 Dunwin Drive, Unit 8, Mississauga, Ontario,
Canada, L5L 1T1
(416) 828-0775



FOLLOW THAT STOCK

A spreadsheet for your stock portfolio

by ERIC VERHEIDEN

The Stock Portfolio Spreadsheet makes it easy for you to keep track of your portfolio of stocks. It lists stocks alphabetically by ticker symbol and includes fields for all pertinent data. Room for 200 stocks is provided (with a 40K Atari). Several summary figures are also provided, and you can add other summary fields if you like. The final report can be listed to either the screen or a printer, or can be stored to disk. Cassette users can store this data by incorporating the changes noted later in the article. Stored reports can also be read and updated with this program.

ENTERING THE PROGRAM

Listing 1 below is the BASIC program. Listing 2 is the relocatable assembly-language sort routine found in the DATA statements at the end of Listing 1, and need not be typed in to use the spreadsheet. After you type in Listing 1, use TYPO to find any typing errors, and SAVE a backup copy for safety.

Eric Verheiden is the author of Vervan's utility programs (CASDUP, CASDIS, FULMAP, DISASM, DISDUP, and DOWNLD) and of the forthcoming Secrets of Atari I/O *(IJG). He holds a Ph.D. from the California Institute of Technology and works for an aerospace firm in Southern California.*

USING THE PROGRAM

The first prompt requests a U for "Update Old File" or C for "Create New File." C erases any previously saved data on the disk. For all prompts, press [RETURN] after responding.

The next prompt reads: ADD, CHANGE, LIST, PRINT, or SAVE.

ADDING ENTRIES

If you choose A in response to the last prompt, you are asked to input a symbol for your new stock. This can be from one to six characters long; different symbols must be supplied for multiple entries of a single stock (e.g., Atari1, Atari2, Atari3).

Next, you are prompted to enter the number of shares, cost per share, and current price of the stock. The next prompt asks you to supply the annual dividend per share. After doing this, you'll be returned to the main menu.

continued on next page

CHANGING AND DELETING ENTRIES

If you choose this option, you must specify which entry is to be changed by entering its symbol. Next, type D to delete the entry or M to modify it. If you choose the latter, you will be presented with the same series of prompts described above (under "Adding Entries"). To change a parameter, type in the new value followed by [RETURN]. If you simply press [RETURN] in response to any prompt, the old value will be retained.

LISTING, PRINTING, AND SAVING

To list your report to the screen, choose L from the main menu. On a 40-column screen, the entries for individual stocks are hard to read, but the summary figures are more legible. The "Print" option produces an 80-column listing on a printer.

Before ending a session with the spreadsheet, the updated listing should be stored to disk with the Save command. The program uses the file name STOCK.DAT. If a larger number of entires are to be added or modified, I recommend that you Save your data several times during the session to protect against its accidental loss.

CASSETTE OPERATION

You should reserve three cassettes for use with the Stock Portfolio Spreadsheet; one for the program, one for the old portfolio listing, and one for the new portfolio listing. Change lines 120 and 990 to read:

```
120 OPEN #3,4,0,"C:"
990 CLOSE #3: OPEN #3,8,0,"C:"
```

To update the old listing, select U from the main menu. After the beep, load the tape containing the old portfolio listing, press PLAY on the recorder, and then press [RETURN] on the computer.

To save the new listing to tape, load a blank cassette into the recorder and press RECORD and PLAY. Then choose S from the main menu. After the double beep, press [RETURN]. Patience is required for these operations, because cassette storage is quite slow.

ADDING NEW SUMMARY LINES

Insert new lines in the program starting with line 1411. Quantities available for manipulation are GSHR (total shares), GCOS (total cost), GVAL (total current value), and GINC (total income). For example, if you call a new quantity GXXX, add these lines:

```
1411 GXXX= expression
1412 S$=STR$(GXXX):GOSUB 1430:
     PRINT #3;" description ";S$
```

This will display the quantity to within two decimal places of accuracy. For three-place accuracy, replace GOSUB 1430 with GOSUB 1460 in line 1412. If you want a percent symbol to appear after the new quantity, append ;"%" to line 1412.

MACHINE-LANGUAGE SORT ROUTINE

Listing 2 provides the source code for the sort routine used in the spreadsheet in Assembler Editor syntax. The routine is relocatable and can sort up to 255 items. Assuming that the sort routine is contained in a string A$ (as in the spreadsheet), the format for calling it is:

X=USR(ADR(A$),ADR(SYMB$),NREC,RECLEN)

Here, SYMB$ is a string that contains NREC fields of RECLEN characters each. NREC and RECLEN must not exceed 255. The ordering of the sorted records is stored at address X, as returned by the sort routine. The sorter produces record indices from zero to NREC−1, so to load the Kth-ordered record you must set IREC=PEEK(X+K−1)+1 and load record IREC.

The routine uses a version of a radix sort and is very fast, since it does not perform actual data movement, but computes an ordering.

Starting with the least-significant bytes of the entries, the program alternately counts the occurrences of each byte ($00–$FF), and then calculates a cumulative partial sum. This is used to take entries in the old ordering and put them into a new ordering, according to the byte under examination. The ordering is finished when this process has been completed for the most significant byte.

The program uses indirect addresses in FR0 for the "old ordering," FR1 for the "new ordering," and FR2 for the accumulated counts. The sign bit of FR0+2 is used as a flag for pass 1 or 2 (plus or minus). FR0+3 contains the current offset into the entries for sorting. Other floating-point registers are used for intermediate results. The entire sort routine is 151 bytes long.

Requires 16K cassette & 24K disk

```
10 REM ** STOCK PORTFOLIO **
15 REM *   SPREAD SHEET    *
20 REM * BY ERIC VERHEIDEN *
25 REM *  ANTIC MAGAZINE   *
30 REM *   FEBUARY, 1984   *
35 REM *********************
40 DIM SYM$(6),ISYM$(6),ISTR$(1),INP$(
90),S$(10),A$(256)
50 DIM SHAR(200),SYMB$(1200),COSTSH(20
0),TCOST(200)
60 DIM PRICESH(200),CURRVAL(200),YIELD
(200),DIV(200),INCOME(200),CHANGE(200)
70 GOSUB 1530
80 PRINT "UPDATE OLD FILE OR CREATE NE
W FILE";:INPUT ISTR$
90 IF ISTR$="C" THEN NREC=0:GOTO 300
100 IF ISTR$<>"U" THEN 80
110 TRAP 290:NREC=-1
120 OPEN #3,4,0,"D:STOCK.DAT"
130 INPUT #3;INP$:IF INP$="" THEN 290
140 NREC=NREC+1
150 IF NREC=0 THEN 130
160 GOSUB 170:GO TO 130
170 S$=INP$(1,5):SHAR(NREC)=VAL(S$)
```

```
180 SYMB$(6*NREC-5)=INP$(7,12)
190 S$=INP$(14,20):COSTSH(NREC)=VAL(S$
)
200 S$=INP$(33,40):PRICESH(NREC)=VAL(S
$)
210 S$=INP$(58,63):DIV(NREC)=VAL(S$)
220 TCOST(NREC)=SHAR(NREC)*COSTSH(NREC
)
230 CURRVAL(NREC)=SHAR(NREC)*PRICESH(N
REC)
240 YIELD(NREC)=100*DIV(NREC)/PRICESH(
NREC)
250 INCOME(NREC)=SHAR(NREC)*DIV(NREC)
260 CHANGE(NREC)=0:IF COSTSH(NREC)=0 T
HEN 280
270 CHANGE(NREC)=100*(PRICESH(NREC)-CO
STSH(NREC))/COSTSH(NREC)
280 RETURN
290 CLOSE #3:IF NREC<0 THEN PRINT "FIL
E NOT FOUND":GO TO 80
300 PRINT "ADD, CHANGE, LIST, PRINT OR
 SAVE";:INPUT ISTR$
310 IF ISTR$="P" THEN 950
320 IF ISTR$="L" THEN 970
330 IF ISTR$="S" THEN 990
340 IREC=NREC+1:IF IREC>200 THEN PRINT
 "TOO MANY RECORDS":STOP
350 IF ISTR$="A" THEN 790
360 IF ISTR$<>"C" THEN 300
370 PRINT "STOCK TO CHANGE";:INPUT ISY
M$
380 IF LEN(ISYM$)>0 THEN IF ASC(ISYM$(
1))=32 THEN ISYM$(1)=ISYM$(2):GO TO 38
0
390 IF LEN(ISYM$)<6 THEN ISYM$(LEN(ISY
M$)+1)=" ":GO TO 390
400 IREC=0
410 IREC=IREC+1:IF IREC>NREC THEN PRIN
T "STOCK NOT FOUND":GO TO 300
420 SYM$=SYMB$(6*IREC-5,6*IREC)
430 IF SYM$<>ISYM$ THEN 410
440 PRINT SHAR(IREC);" SHARES ";SYM$
450 S$=STR$(COSTSH(IREC)):GOSUB 1460:P
RINT "COST PER SHARE $";S$
460 S$=STR$(TCOST(IREC)):GOSUB 1460:PR
INT "TOTAL COST $";S$
470 S$=STR$(PRICESH(IREC)):GOSUB 1460:
PRINT "CURRENT PRICE PER SHARE $";S$
480 S$=STR$(CURRVAL(IREC)):GOSUB 1460:
PRINT "CURRENT VALUE $";S$
490 S$=STR$(YIELD(IREC)):GOSUB 1460:PR
INT "CURRENT YIELD ";S$;"%"
500 S$=STR$(DIV(IREC)):GOSUB 1460:PRIN
T "DIVIDEND PER SHARE $";S$
510 S$=STR$(INCOME(IREC)):GOSUB 1460:P
RINT "INCOME $";S$
520 S$=STR$(CHANGE(IREC)):GOSUB 1460:P
RINT "NET CHANGE ";S$;"%"
530 TRAP 300:PRINT "DELETE OR MODIFY E
NTRY";:INPUT ISTR$
540 IF ISTR$="M" THEN 640
550 IF ISTR$<>"D" THEN 530
560 IF IREC=NREC THEN 630
570 SYMB$(6*IREC-5)=SYMB$(6*IREC+1)
580 FOR I=IREC TO NREC-1
590 SHAR(I)=SHAR(I+1):COSTSH(I)=COSTSH
(I+1)
600 TCOST(I)=TCOST(I+1):PRICESH(I)=PRI
CESH(I+1)
610 CURRVAL(I)=CURRVAL(I+1):YIELD(I)=Y
IELD(I+1)
620 DIV(I)=DIV(I+1):INCOME(I)=INCOME(I
+1):CHANGE(I)=CHANGE(I+1):NEXT I
630 NREC=NREC-1:GO TO 300
640 NREC=NREC+1
650 PRINT "NEW SYMBOL";:SYMB$(6*NREC-5
)=SYMB$(6*IREC-5,6*IREC):INPUT SYM$
660 IF LEN(SYM$)>0 THEN IF ASC(SYM$(1)
)=32 THEN SYM$(1)=SYM$(2):GO TO 660
670 IF LEN(SYM$)=0 THEN 700
680 IF LEN(SYM$)<6 THEN SYM$(LEN(SYM$)
+1)=" ":GOTO 680
690 SYMB$(6*NREC-5)=SYM$
700 PRINT "NEW SHARES";:SHAR(NREC)=SHA
R(IREC):INPUT INP$
710 IF LEN(INP$)>0 THEN SHAR(NREC)=VAL
(INP$)
720 PRINT "NEW COST";:COSTSH(NREC)=COS
TSH(IREC):INPUT INP$
730 IF LEN(INP$)>0 THEN COSTSH(NREC)=V
AL(INP$)
740 PRINT "NEW CURRENT PRICE";:PRICESH
(NREC)=PRICESH(IREC):INPUT INP$
750 IF LEN(INP$)>0 THEN PRICESH(NREC)=
VAL(INP$)
760 PRINT "NEW DIVIDEND";:DIV(NREC)=DI
V(IREC):INPUT INP$
770 IF LEN(INP$)>0 THEN DIV(NREC)=VAL(
INP$)
780 GOSUB 220:GO TO 560
790 NREC=NREC+1:IREC=NREC
800 PRINT "SYMBOL";:INPUT SYM$
810 IF LEN(SYM$)>0 THEN IF ASC(SYM$(1)
)=32 THEN SYM$(1)=SYM$(2):GO TO 810
820 IF LEN(SYM$)<6 THEN SYM$(LEN(SYM$)
+1)=" ":GOTO 820
830 SYMB$(6*IREC-5)=SYM$
840 PRINT "SHARES";:INPUT X:SHAR(IREC)
=X
850 PRINT "COST";:INPUT X:COSTSH(IREC)
=X
860 PRINT "CURRENT PRICE";:INPUT X:PRI
CESH(IREC)=X
870 PRINT "DIVIDEND";:INPUT X:DIV(IREC
)=X
880 TCOST(IREC)=SHAR(IREC)*COSTSH(IREC
)
890 CURRVAL(IREC)=SHAR(IREC)*PRICESH(I
REC)
900 YIELD(IREC)=100*DIV(IREC)/PRICESH(
IREC)
910 INCOME(IREC)=SHAR(IREC)*DIV(IREC)
920 CHANGE(IREC)=0:IF COSTSH(IREC)=0 T
HEN 940
930 CHANGE(IREC)=100*(PRICESH(IREC)-CO
STSH(IREC))/COSTSH(IREC)
940 GO TO 300
```

continued on page 35

NOVIN™

ANNOUNCES

ANTICA-4 T.M.



Now you can take complete advantage of the mode 4 color character graphics capabilities of your Atari 48K 800 or 1200XL disc-based computer! This system provides total control over character generation and graphics mixing, with such features as:

- AUTOMATIC CHARACTER FILE: ANTICA-4 lets you design character graphics without having to set up or catalogue your character-file, knows when you need a new character or simply an edit, and makes characters keyboard assignable.
- AUTOMATIC MULTI-CHARACTER MATTING: This feature lets you perfectly matte moving characters over a moving background, for full character and graphics mixing.
- MULTI-CHARACTER PICTURE ANIMATOR: Name your moving pictures, put them in motion, and watch your creations automatically matte themselves over a full color animated background. Your animations edit easily and automatically.
- MACRO-MICRO CONTROL: The field can be any shape and several times larger than the screen. Work your field with character resolution in the Macro mode, or build your characters with an enlarged view in the Micro mode.
- FINE-LINE DRAWING: Make fine-line drawings over complex backgrounds in any of your colors while ANTICA-4 continuously and automatically updates your character file.
- INTER-MODAL CONSISTENCY: Regardless of the character source, ANTICA-4 lets you easily switch between the editor, animation and building modes without loss of system integrity.
- PLUS: Full function matting editor.
- PLUS: Five speed fine-scrolling field.
- PLUS: Full color and luminosity control, with proportional dimming.
- PLUS: DOS compatibility.
- AND MUCH MORE: Multi-character field-fill, automatic field or cursor-wipe replacement of characters or backgrounds, combined characters, the list goes on and on. You have an incredibly creative system at your command. It's simply a treat to watch it work!

NOVIN is proud to offer this fantastic package for $49.95. ANTICA-4 arrives with a full character set and file of animated pictures. Ask for it at your computer store or order direct from us toll-free.

PO Box 22889
Seattle, Washington 98122

NOVIN™

1-206-323-8270
Outside Washington call toll-free
1-800-422-5525

Visa and Mastercharge accepted. Atari® is a registered trademark of Atari, Inc.

FOLLOW THAT STOCK continued from page 33

```
950 CLOSE #3:OPEN #3,8,0,"P:"
960 GOTO 1000
970 CLOSE #3:OPEN #3,8,0,"E:"
980 GO TO 1000
990 CLOSE #3:OPEN #3,8,0,"D:STOCK.DAT"
1000 PRINT #3;" #SHR SYMBOL COST/SH TO
TAL COST PRICE/SH ";
1010 PRINT #3;"CURR VALUE %CHG  DIV/SH
   INCOME  YIELD"
1020 IF NREC=0 THEN 300
1030 X=USR(ADR(A$),ADR(SYMB$),NREC,6)
1040 GSHR=0:GCOS=0:GVAL=0:GINC=0:FOR K
=1 TO NREC
1050 IREC=PEEK(X+K-1)+1
1060 S$=STR$(SHAR(IREC)):GSHR=GSHR+SHA
R(IREC):INP$(1)="        "
1070 INP$(6-LEN(S$))=S$:INP$(6)=" "
1080 INP$(7)=SYMB$(6*IREC-5,6*IREC)
1090 S$=STR$(COSTSH(IREC)):JOLD=10
1100 J=17:GOSUB 1440
1110 S$=STR$(TCOST(IREC)):GCOS=GCOS+TC
OST(IREC)
1120 J=29:GOSUB 1450
1130 S$=STR$(PRICESH(IREC))
1140 J=36:GOSUB 1440
1150 S$=STR$(CURRVAL(IREC)):GVAL=GVAL+
CURRVAL(IREC)
1160 J=48:GOSUB 1450
1170 J=CHANGE(IREC):S$=STR$(J+SGN(J)/2
):GOSUB 1460
1180 S$(I)="%":INP$(51)="       ":INP$(5
7-LEN(S$))=S$:JOLD=54
1190 S$=STR$(DIV(IREC))
1200 J=60:GOSUB 1440
1210 S$=STR$(INCOME(IREC)):GINC=GINC+I
NCOME(IREC)
1220 J=71:GOSUB 1450
1230 S$=STR$(YIELD(IREC))
1240 J=77:GOSUB 1440:INP$(80)="%"
1250 PRINT #3;INP$:NEXT K
1260 PRINT #3:PRINT #3;"GRAND TOTAL SH
ARES   ";GSHR
1270 S$=STR$(GCOS):GOSUB 1430:PRINT #3
;"GRAND TOTAL COST   $";S$
1280 S$=STR$(GVAL):GOSUB 1430:PRINT #3
;"GRAND TOTAL VALUE  $";S$
1290 S$=STR$(GINC):GOSUB 1430:PRINT #3
;"GRAND TOTAL INCOME $";S$
1300 GDEL=100*(GVAL-GCOS)/GCOS
1310 S$=STR$(GDEL):GOSUB 1460:PRINT #3
;"AVERAGE INCREASE ";S$;"%"
1320 GYIE=100*GINC/GVAL
1330 S$=STR$(GYIE):GOSUB 1460:PRINT #3
;"AVERAGE YIELD    ";S$;"%"
1340 GYIE=100*GINC/GCOS
1350 S$=STR$(GYIE):GOSUB 1460:PRINT #3
;"INCOME/COST       ";S$;"%"
1360 GYIE=GINC/12
1370 S$=STR$(GYIE):GOSUB 1430:PRINT #3
;"INCOME PER MONTH $";S$
1380 GYIE=GINC/52
1390 S$=STR$(GYIE):GOSUB 1430:PRINT #3
;"INCOME PER WEEK  $";S$
1400 GYIE=GINC/2080
1410 S$=STR$(GYIE):GOSUB 1430:PRINT #3
;"INCOME PER HOUR  $";S$
1420 CLOSE #3:GO TO 300
1430 GOSUB 1460:S$=S$(1,I+2):RETURN
1440 GOSUB 1450:JOLD=JOLD+1:RETURN
1450 GOSUB 1460:INP$(JOLD+3)="
   ":INP$(J+1-I)=S$:JOLD=J:RETURN
1460 FOR I=1 TO 10:IF I>LEN(S$) THEN S
$(I)=".":GO TO 1490
1470 IF ASC(S$(I))=46 THEN 1490
1480 NEXT I:GO TO 1510
1490 IF LEN(S$)=I THEN S$(I+1)="00"
1500 IF LEN(S$)=I+1 THEN S$(I+2)="0"
1510 IF LEN(S$)>I+3 THEN S$=S$(1,I+3)
1520 RETURN
1530 DATA 104,104,133,244,104,133,243,
104
1540 DATA 104,133,242,104,104,133,216,
133
1550 DATA 215,198,215,166,89,202,134,2
13
1560 DATA 202,134,231,202,134,225,160,
0
1570 DATA 132,212,132,230,132,224,132,
214
1580 DATA 152,145,212,200,208,250,24,1
02
1590 DATA 214,160,0,152,72,36,214,16
1600 DATA 3,24,113,224,170,104,145,224
1610 DATA 138,200,208,240,162,0,138,16
8
1620 DATA 177,212,72,133,217,169,0,133
1630 DATA 219,160,7,10,38,219,6,217
1640 DATA 144,7,24,101,216,144,2,230
1650 DATA 219,136,16,239,24,101,243,13
3
1660 DATA 218,165,219,101,244,133,219,
164
1670 DATA 215,177,218,168,177,224,72,2
4
1680 DATA 105,1,145,224,104,168,104,14
5
1690 DATA 230,232,228,242,144,192,36,2
14
1700 DATA 16,165,165,213,166,231,134,2
13
1710 DATA 133,231,198,215,16,152,96
1720 FOR I=1 TO 151
1730 READ X:A$(I)=CHR$(X):NEXT I
1740 RETURN
```

TYPO TABLE

Variable checksum = 1286082

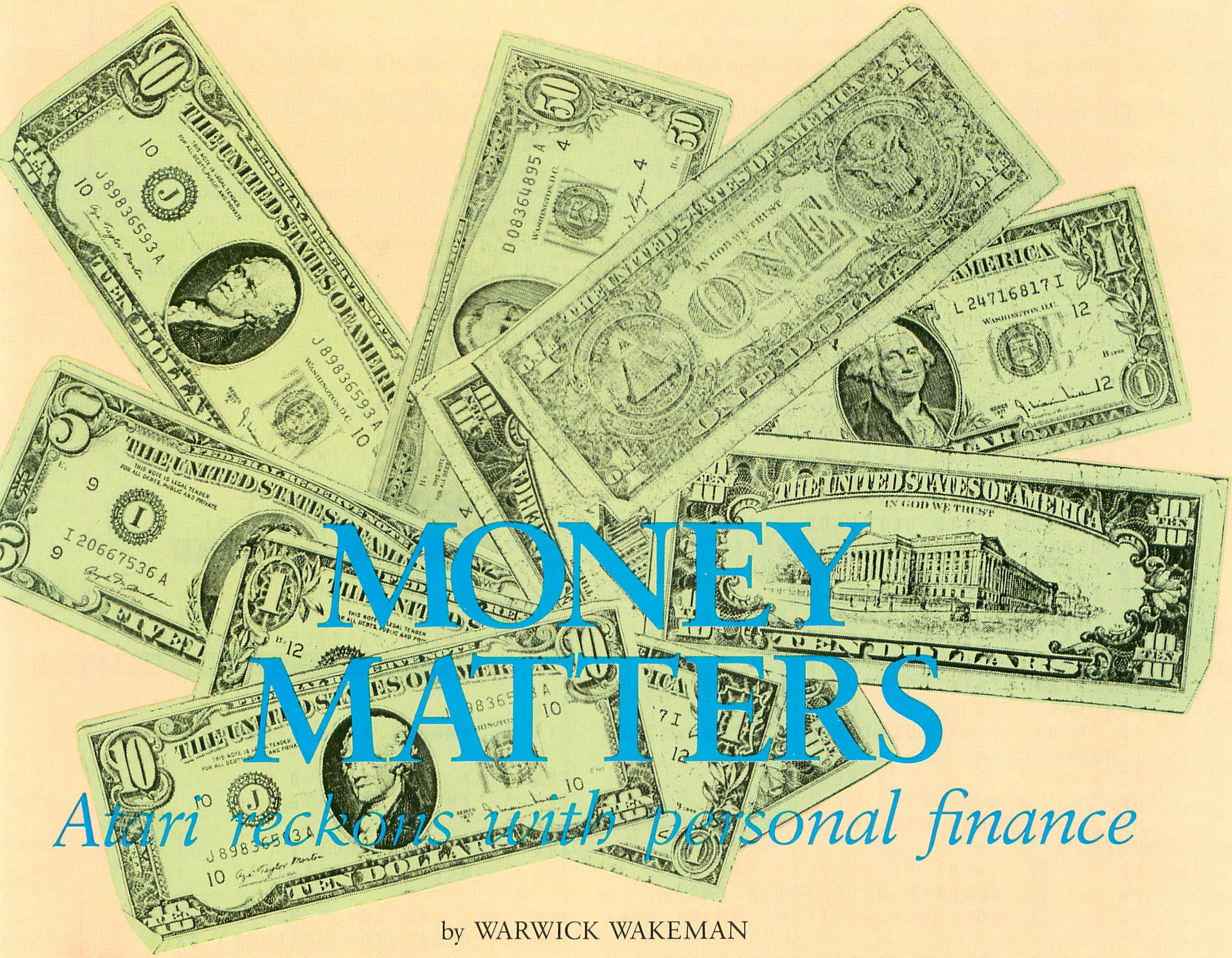
Line num range	Code	Length
10 - 90	XM	446
100 - 210	SX	380
220 - 330	YK	327
340 - 450	GB	439
460 - 570	JW	484
580 - 690	YA	516
700 - 810	VT	408
820 - 930	AH	361
940 - 1050	SH	449

continued on next page

```
1060  - 1170    SQ   372
1180  - 1290    SI   459
1300  - 1410    XH   457
1420  - 1530    WR   423
1540  - 1650    ZL   414
1660  - 1740    UD   262
```

Listing 2

```
1000          *=$600
1010 ;
1020 SAVMSC  =        $58
1030 FR0     =        $D4
1040 FRE     =        $DA
1050 FR2     =        $E0
1060 FR1     =        $E6
1070 CIX     =        $F2
1080 INBUFF  =        $F3
1090 ;
1100 ; X=USR(1536,ADR(A$),N,L)
1110 ;
1120 ; A$ - ENTRIES TO BE SORTED
1130 ; N  - NUMBER OF ENTRIES
1140 ; L  - LENGTH OF ENTRIES
1150 ; X  - ADDRESS OF POINTER TABLE
1160 ;
1170         PLA
1180         PLA
1190         STA      INBUFF+1
1200         PLA
1210         STA      INBUFF
1220         PLA
1230         PLA
1240         STA      CIX
1250         PLA
1260         PLA
1270         STA      FR0+4
1280         STA      FR0+3
1290         DEC      FR0+3
1300         LDX      SAVMSC+1
1310         DEX
1320         STX      FR0+1
1330         DEX
1340         STX      FR1+1
1350         DEX
1360         STX      FR2+1
1370         LDY      #0
1380         STY      FR0
1390         STY      FR1
1400         STY      FR2
1410         STY      FR0+2
1420 PASS0   TYA
1430         STA      (FR0),Y
1440         INY
1450         BNE      PASS0
1460 ;
1470 PASS1   CLC
1480 PASS2   ROR      FR0+2
1490         LDY      #0
1500         TYA
1510 SET1    PHA
1520         BIT      FR0+2
1530         BPL      NADD
1540 ;
1550         CLC
1560         ADC      (FR2),Y
1570 NADD    TAX
1580         PLA
1590         STA      (FR2),Y
1600         TXA
1610         INY
1620         BNE      SET1
1630 ;
1640         LDX      #0
1650 ORDR    TXA
1660         TAY
1670         LDA      (FR0),Y
1680         PHA
1690         STA      FR0+5
1700         LDA      #0
1710         STA      FRE+1
1720         LDY      #$07
1730 SHFT    ASL      A
1740         ROL      FRE+1
1750         ASL      FR0+5
1760         BCC      NEXB
1770 ;
1780         CLC
1790         ADC      FR0+4
1800         BCC      NEXB
1810 ;
1820         INC      FRE+1
1830 NEXB    DEY
1840         BPL      SHFT
1850 ;
1860         CLC
1870         ADC      INBUFF
1880         STA      FRE
1890         LDA      FRE+1
1900         ADC      INBUFF+1
1910         STA      FRE+1
1920         LDY      FR0+3
1930         LDA      (FRE),Y
1940         TAY
1950         LDA      (FR2),Y
1960         PHA
1970         CLC
1980         ADC      #1
1990         STA      (FR2),Y
2000         PLA
2010         TAY
2020         PLA
2030         STA      (FR1),Y
2040         INX
2050         CPX      CIX
2060         BCC      ORDR
2070 ;
2080         BIT      FR0+2
2090         BPL      PASS2
2100 ;
2110         LDA      FR0+1
2120         LDX      FR1+1
2130         STX      FR0+1
2140         STA      FR1+1
2150         DEC      FR0+3
2160         BPL      PASS1
2170         RTS
2180 ;
2190         .END
```



MONEY MATTERS

Atari reckons with personal finance

by WARWICK WAKEMAN

The recent economic recession has brought home to the average person how important it is to be aware of personal finance. The issues of uncertain investment returns, inflation and unreasonable mortgage rates are complex, and most of us are not financial professionals with access to the tables and charts that can ease the pain of complicated mathematical calculations. If you're like me, you lapse into a funk when brought face to face with an algebraic equation involving "principal, interest and time."

Knowing that my Atari 800 would not be the least bit fazed by such a confrontation, I collected a series of financial equations, wrote a few subroutines, added a touch of color, and came up with the following menu-driven program. While it won't solve your financial dilemmas, it might make your struggles less burdensome.

PROGRAM DESCRIPTION

The program opens with an invitation to choose one of several options: "WHAT WILL MY INVESTMENT BE WORTH?"; "VALUE OF REGULAR SAVINGS"; "ANNUITY INCOME"; and so on. One menu choice leads the user to a brief description of how each option can be used. For example, "DEVALUATION OF THE $ BY INFLATION" not only determines the effect of inflation on your savings, it also can calculate the depreciation of a piece of property, such as the family car. By using this option in conjunction with "GROWTH RATE OF MONEY," you can obtain an accurate picture of the actual increase (or decrease) in the value of an investment over a given period of time.

The program is written in very basic BASIC. It is simply a series of subroutines tied together by a menu. The calculating routines (lines 100, 300, 500, 700, 900, and 1100) in the program can be changed to meet your specific requirements. Osborne/McGraw-Hill's *Some Common BASIC Programs (Atari Edition)* contains a number of programs you may find useful. TAB Books' *The Most Popular Subroutines in BASIC* is also crammed full of usable formulas.

TAKE APART

Lines 0–25: Give the program a title.
Lines 30–95: Menu information; requests user input and

continued on next page

directs the program to the chosen menu option. The program error traps each INPUT line (see lines 95 and 120, for example).

Lines 100–195: This section does the work involved in the first menu choice. After requesting user input, it computes the figures you need (lines 155, 160 and 175) and prints the answer on the screen (lines 170 and 180).

Line 190: This subroutine (GOSUB 3000) asks the user to stay with this choice, ask for further directions, or return to the main menu. All of the routines are set up in virtually the same way.

Line 2000: This final menu selection (menu choice 7: line 85) provides you with details about how to use each option, and suggests further applications. If you don't want this section in your program, eliminate line 85, omit 2000 from line 95, and eliminate lines 2000 to 2285.

Lines 3000–3135: After each calculation, these two subroutines lead the user back into the program to continue with the function, get more information or return to the main menu. If you choose to eliminate the "FURTHER DETAIL" routine (lines 2000–2285), then lines 3010, 3030, 3110 and 3130 should be omitted and lines 3015, 3035 and 3135 changed.

Requires 16K RAM

```
1 REM *** MONEY MATTERS ****
2 REM * BY WARWICK WAKEMAN *
3 REM *    ANTIC MAGAZINE   *
4 REM *    FEBUARY, 1984    *
5 REM **********************
8 GRAPHICS 1+16:SETCOLOR 2,4,4
9 POSITION 6,5:? #6;"FINANCIAL"
10 POSITION 4,7:? #6;"CALCULATIONS"
15 POSITION 9,12:? #6;"BY"
20 POSITION 6,14:? #6;"WARWICK!"
25 FOR X=1 TO 75 STEP 0.05:NEXT X:?
30 GRAPHICS 0:SETCOLOR 2,4,4:? :? :?
35 PRINT "THIS PROGRAM WILL COMPUTE FINANCIAL"
40 PRINT "  CALCULATIONS FOR INVESTMENT AND"
45 ? "              BORROWING"
50 ? :? :? "CHOOSE:":?
55 ? " 1. WHAT WILL MY INVESTMENT BE WORTH?"
60 ? " 2. VALUE OF REGULAR SAVINGS"
65 ? " 3. ANNUITY OR PENSION INCOME"
70 ? " 4. HOW MUCH DO I OWE ON MY LOAN?"
75 ? " 5. GROWTH OF INTEREST BEARING INVEST"
80 ? " 6. DEVALUATION OF THE $ BY INFLATION"
85 ? " 7. FURTHER DETAIL ON 1  THROUGH 7."
90 TRAP 50:INPUT X
95 ON X GOTO 100,300,500,700,900,1100,2000
100 GRAPHICS 0:SETCOLOR 2,13,6:? :?
105 ? "    FUTURE WORTH OF AN INVESTMENT"
110 ?
115 ? "ORIGINAL INVESTMENT              $";
120 TRAP 115:INPUT OA
125 ? "INTEREST RATE                    %";
130 TRAP 125:INPUT I
135 ? "COMPOUNDING PERIODS PER YEAR     #";
140 TRAP 135:INPUT N
145 ? "TOTAL YEARS                      #";
150 TRAP 145:INPUT T
155 I=I/N*0.01
160 FW=OA*(1+I)^(N*T)
165 ?
170 ? "FUTURE WORTH = $";
175 ? INT(FW*100+0.6)*0.01
180 ? "AT END OF ";T;" YEARS"
185 ?
190 GOSUB 3000
195 GOTO 110
300 GRAPHICS 0:SETCOLOR 2,2,4:? :?
305 ? "FUTURE VALUE OF REGULAR SAVINGS"
310 ?
315 ? "AMOUNT EACH DEPOSIT              $";
320 TRAP 315:INPUT D
325 ? "INTEREST RATE                    %";
330 TRAP 325:INPUT I
335 ? "DEPOSITS PER YEAR                #";
340 TRAP 335:INPUT N
345 ? "NUMBER OF YEARS TO SAVE          #";
350 TRAP 345:INPUT T
355 I=I/N*0.01
360 FV=D*((1+I)^(N*T)-1)/I
365 ?
370 ? "FUTURE VALUE = $";
375 ? INT(FV*100+0.6)*0.01
380 ? "AT END OF ";T;" YEARS"
385 ?
390 GOSUB 3000
395 GOTO 310
500 GRAPHICS 0:SETCOLOR 2,5,4:? :?
505 ? "AMOUNT REQ'D FOR ANNUITY OR PENSION"
510 ?
515 ? "AMOUNT OF PLANNED WITHDRAWALS $";
520 TRAP 515:INPUT PW
525 ? "EXPECTED INTEREST RATE           %";
530 TRAP 525:INPUT I
535 ? "PLANNED WITHDRAWALS PER YEAR     #";
540 TRAP 535:INPUT N
```

```
545 ? "NUMBER OF YEARS TO MATURITY    #
";
550 TRAP 545:INPUT T
555 I=I*0.01
560 AR=PW*N/I*(1-1/((1+I/N)^(N*T)))
565 ?
570 ? "MINIMUM INVESTMENT REQ'D = $";
575 ? INT(100*AR+0.6)*0.01
580 ? "FOR ";N*T;" WITHDRAWALS IN ";T
;" YEARS"
585 ?
590 GOSUB 3000
595 GOTO 510
700 GRAPHICS 0:SETCOLOR 2,1,3:? :?
705 ? "       BALANCE OWING ON A LOAN"
710 ?
715 ? "REGULAR PAYMENT                 $
";
720 TRAP 715:INPUT RP
725 ? "ORIGINAL AMOUNT BORROWED        $
";
730 TRAP 725:INPUT P
735 ? "NUMBER OF PAYMENTS PER YEAR     #
";
740 TRAP 735:INPUT N
745 ? "ANNUAL INTEREST RATE            %
";
750 TRAP 745:INPUT I
755 ? "TOTAL NUMBER OF PAYMENTS MADE #
";
760 TRAP 755:INPUT T
765 I=I/100
770 J=I/N
775 BO=P*(1+J)^T-RP*((1+J)^T-(1+J))/J-
(1*RP)
780 ?
785 ? "BALANCE OWING = $";
790 ? INT(BO*100+0.6)*0.01
795 ? "AFTER ";T;" PAYMENTS"
800 ?
805 GOSUB 3000
810 GOTO 710
900 GRAPHICS 0:SETCOLOR 2,3,2:? :?
905 ? "       GROWTH OF MONEY"
910 ?
915 ?
920 ? "INITIAL AMOUNT        $";
925 TRAP 920:INPUT P
930 ? "INTEREST RATE         %";
935 TRAP 930:INPUT R
940 LET N=1
945 LET A=P*(1+R/100)^N
950 ?
955 ? "YEAR    = #";N
960 ?
965 ? "AMOUNT  = $";A
970 LET N=N+1
975 ?
980 ? "FOR ADDITIONAL YEARS, HIT 1"
985 ? "TO CONTINUE PROGRAM, HIT 2"
990 TRAP 980:INPUT X
995 IF X=1 THEN 945
1000 IF X=2 THEN GOSUB 3000
1005 GOTO 915
1100 GRAPHICS 0:SETCOLOR 2,14,2:? :?
1105 ?
1110 ? " $ DEVALUATION AT RATE OF INFL
ATION"
1115 ?
1120 ? "ORIGINAL VALUE                $"
;
1125 TRAP 1120:INPUT OV
1130 ? "INFLATION RATE                %"
;
1135 TRAP 1130:INPUT I
1140 ? "YEARS                         #"
;
1145 TRAP 1140:INPUT T
1150 ?
1155 DV=INT(100*OV*(1-I/100)^T+0.6)*0.
01
1160 ? "DEVALUATION = $";DV
1165 ? "AT END OF ";T;" YEARS"
1170 ? :?
1175 ? "FOR ADDITIONAL YEARS, HIT 1"
1180 ? "TO CONTINUE PROGRAM, HIT 2"
1185 TRAP 1175:INPUT X
1190 IF X=1 THEN 1210
1195 IF X=2 THEN GOSUB 3000
1200 GOTO 1120
1205 ? :?
1210 ? "ENTER NUMBER OF ADDITIONAL YEA
RS - ";
1215 TRAP 1210:INPUT N
1220 ? "ENTER INFLATION RATE FOR ADDIT
IONAL YEARS - %";
1225 TRAP 1220:INPUT I
1230 DV=INT(100*DV*(1-I/100)^N+0.6)*0.
01
1235 LET T=T+N
1240 ? :?
1245 ? "DEVALUATION OF $";OV;" = $";DV
1250 ? "AT END OF ";T;" YEARS"
1255 ?
1260 GOTO 1175
2000 GRAPHICS 0:SETCOLOR 2,10,1:? :?
2005 ? "   THIS PART OF THE PROGRAM WIL
L          GIVE FURTHER DETAILS ON THE
VARIOUS    CALCULATIONS":? :?
2010 ? "CHOOSE:"
2015 ? " 1. WHAT WILL MY INVESTMENT BE
 WORTH?"
2020 ? " 2. VALUE OF REGULAR SAVINGS"
2025 ? " 3. ANNUITY OR PENSION INCOME"
2030 ? " 4. HOW MUCH DO I OWE ON MY LO
AN?"
2035 ? " 5. GROWTH OF INTEREST BEARING
 INVEST"
2040 ? " 6. DEVALUATION OF THE $ BY IN
FLATION"
2045 ? " 7. BACK TO MAIN MENU"
2050 TRAP 2005:INPUT Z
2055 ON Z GOTO 2100,2140,2170,2205,223
5,2260,30
2100 GRAPHICS 0:COLOR 1:SETCOLOR 2,13,
6
```

continued on page 41


"Exodus: Ultima III, with a superior plot to match its superior gaming system, is a great game . . . it sets new standards for fantasy gaming state of the art."
Softline, November/December 1983
"Caverns of Callisto is a very challenging and enjoyable arcade game. I hope Origin Systems can continue to provide products of such quality."
Core, December 1983
CAVERNS OF CALLISTO
Created by Chuckles
$34.95
Apple and Atari.
EXODUS
ULTIMA III
Created by Lord British
$59.95
Apple, Atari, Com 64 and IBM
Apple version MOCKINGBOARD enhanced.
ORIGIN SYSTEMS INC.
P. O. Box 99 N. Andover, MA 01845
(617) 681-0609
"Exodus: Ultima III . . . is fun and exciting to play and constantly intriguing."
Softalk, November 1983
LOUBET
Apple, Atari, Com64, and IBM are trademarks of Apple Inc., Atari Inc., Commodore Business Machines, and IBM, respectively.
Ultima and Lord British are trademarks of Richard Garriott. Copyright 1983 by Origin Systems, Inc.

```
2105 ? :? "      FUTURE WORTH OF AN INVESTMENT":?
2110 ? " This function will compute the worth of an investment at a specified futuretime. The original investmen";
2115 ? "t can be asavings account, a term deposit, bondsor a retirement fund."
2120 ? :? " It may also be used to estimate the  future value of an antique or a piece of real estate. ";
2125 ? " Just enter what you  estimate the appreciation rate is in place of the INTEREST RATE."
2130 GOSUB 3100
2135 GOTO 100
2140 GRAPHICS 0:COLOR 1:SETCOLOR 2,2,4:? :?
2145 ? :? "     FUTURE VALUE OF REGULAR SAVINGS"
2150 ? "This function will compute the future value of regular equal amounts deposited in an interest ";
2155 ? "bearing investment  such as a savings account, insurance  policy or retirement savings plan."
2160 GOSUB 3100
2165 GOTO 300
2170 GRAPHICS 0:COLOR 1:SETCOLOR 2,5,4:? :?
2175 ? :? " MINIMUM ANNUITY INVESTMENT REQUIRED"
2180 ? "This function will compute the amount required to provide regular income    over a specified period of";
2185 ? " time such  as pension payments. It can also be  used to determine the amount required to provide";
2190 ? " living expenses while attending college or university."
2195 GOSUB 3100
2200 GOTO 500
2205 GRAPHICS 0:COLOR 1:SETCOLOR 2,1,3:? :?
2210 ? :? "     BALANCE REMAINING ON A LOAN"
2215 ? "This function will compute the balanceowing on a loan after a specified number of payments such as a";
2220 ? " bank loan or a mortgage."
2225 GOSUB 3100
2230 GOTO 700
2235 GRAPHICS 0:COLOR 1:SETCOLOR 2,3,2:? :?
2240 ? :? "     GROWTH RATE OF MONEY"
2245 ? "This function will show how money    grows year by year when placed in an interest bearing investment."
2250 GOSUB 3100
2255 GOTO 900
2260 GRAPHICS 0:COLOR 1:SETCOLOR 2,14,2:? :?
2265 ? :? "    $ DEVALUATION AT RATE OF INFLATION"
2270 ? "This function will compute the effect on inflation on the doller.  It can   also be used to calculate ";
2275 ? "depreciationon a piece of equipment.  Just enter  the depreciation rate in place of the INFLATION RATE."
2280 GOSUB 3100
2285 GOTO 1100
3000 ? :? :? "CHOOSE:"
3005 ? "1. CONTINUE WITH THIS FUNCTION?"
3010 ? "2. FURTHER DATA ON CALCULATIONS?"
3015 ? "3. BACK TO MAIN MENU?"
3020 TRAP 3000:INPUT Y
3025 IF Y=1 THEN RETURN
3030 IF Y=2 THEN 2000
3035 IF Y=3 THEN 30
3100 ? :? "CHOOSE:"
3105 ? "1. TO USE THIS FUNCTION"
3110 ? "2. FURTHER DATA ON CALCULATIONS"
3115 ? "3. BACK TO MAIN MENU"
3120 TRAP 3040:INPUT A
3125 IF A=1 THEN RETURN
3130 IF A=2 THEN 2000
3135 IF A=3 THEN 30
```

TYPO TABLE

Variable checksum = 219221

Line num range			Code	Length
1	-	30	UO	444
35	-	90	PH	446
95	-	150	NB	382
155	-	310	UZ	262
315	-	370	PB	313
375	-	530	BY	303
535	-	590	PP	327
595	-	750	PA	329
755	-	810	IX	278
900	-	955	WC	256
960	-	1105	HF	247
1110	-	1165	JT	332
1170	-	1225	JY	297
1230	-	2020	YB	406
2025	-	2115	FD	563
2120	-	2155	EM	528
2160	-	2210	HO	532
2215	-	2270	TZ	609
2275	-	3100	AC	365
3105	-	3135	HN	179

TYPO

Type your program once

by BILL WILKINSON

TYPO is designed to help you find the typing errors you make when entering the BASIC programs published in ANTIC. TYPO generates a table of values called a "Typo Table," which includes a variable checksum, line-number ranges, codes and lengths. Compare your table with the appropriate Typo Table in ANTIC. (A different table accompanies each BASIC listing.) If the two are not identical, you've probably made a "typo" that needs to be corrected.

HOW TO USE TYPO

These instructions apply when you use TYPO to test other programs. See the next section for instructions on how to use TYPO to check itself.

1. Enter the program listing exactly as shown.
2. LIST this program to disk (LIST "D:TYPO.LST") or cassette (via LIST "C:"). When using a cassette, reserve an entire blank cassette for TYPO.

"TYPO" first appeared in Volume 1, Number 3 of ANTIC, and was reprinted in Volume 2, Number 1. Now that our listings are set in a monospaced font, we reprint it here as a service to our many new readers. Bill Wilkinson, Vice-President and Technical Director of Optimized Systems Software, Inc., was one of the original designers of Atari BASIC. —ANTIC ED

3. Type NEW to clear memory.
4. Type in a program from ANTIC.
5. LIST this program to disk (LIST "D:NAME") or cassette (LIST"C:"). Type NEW and reenter the program (ENTER "D:NAME" or ENTER "C:").
6. Append the TYPO program onto the end of the program from the disk (ENTER "D:TYPO.LST") or cassette (ENTER "C:").
7. Type GOTO 32000. You will be prompted for "File for output." Type [S] [RETURN] for screen output or [P] [RETURN] to send output to the printer. Your Typo Table will then be printed to the appropriate device. Compare your table with the one published with the BASIC listing. If they agree, you're finished and the program should run.
8. If any values in the two tables do not agree, note the variable-checksum value and keep it handy.
9. Examine the lines which have codes and/or lengths that disagree. Correct any errors.
10. *If and only if* the variable checksum you noted agrees with the one in the magazine, return to Step 7 (above) and try again.
11. If the variable checksums do not agree, you *must* go to Step 5 (above) and perform the listing and reentering ritual! You may skip Step 6, however, since (presumably) you have already merged the two programs with LIST and ENTER.

HOW TO USE TYPO ON ITSELF

1. Follow Steps 1–3 above.
2. Load the saved TYPO (ENTER "D:TYPO.LST" or ENTER "C:").
3. Change 32000 to 32500 in lines 32180 and 32200. This alteration applies *only* when you use TYPO to check itself.
4. Go to Step 7 (above).

WHAT TYPO IS TELLING YOU

This program is fussy! It counts every period, comma, and space. It also cares about the order in which you type in program lines! The order in which variable names are stored depends on the order in which the lines were typed. If this order is altered, the values of the tokens and subsequent checksums will also be changed.

The variable checksum is used to correct for this by producing a checksum that depends on the order in which variables are stored. If your checksum doesn't agree, you have either entered lines in the wrong order or misspelled a variable name. In either case, you *must* correct your error(s) and then go through the LIST/NEW/ENTER sequence to ensure that the variables are in the correct order.

The length shown is the number of bytes in the tokenized program that are

encountered by TYPO within the indicated line-number range. The two-letter code is essentially a checksum of "length" bytes within that same range. If the length is correct but the code is wrong, you may have made a spelling or punctuation error. Be careful: since all keywords and operators (including two-character operators such as "<>") are tokenized as one byte, the length may stay the same even though you type SETCOLOR for CLR. You may use abbreviations for keywords (see "Appendix A" of the *BASIC Reference Manual*) as long as the LISTed result conforms to the magazine listing.

If the length bytes disagree, you have added or deleted characters. If nothing obvious shows, pay special attention to characters in quoted strings and/or REM statements. It is easy to omit a space or punctuation in a REMark, thinking that "REMs don't matter." But to TYPO, they do.

```
32000 REM Type Your Program Once -- "TYPO"
32100 CLR :DIM Q$(20):QF=7:CLOSE #QF:? "File for output ";
32110 INPUT Q$:OPEN #QF,12,0,Q$:QREM=0
32130 QCNT=1:FOR QADDR=PEEK(130)+256*PEEK(131) TO PEEK(132)+256*PEEK(133)-1
32140 QSUM=QSUM+PEEK(QADDR)*QCNT:QCNT=QCNT+1:NEXT QADDR
32150 ? #QF;"Variable checksum = ";QSUM:? #QF
32160 QADDR=PEEK(136)+256*PEEK(137):? #QF;"    Line num range      Code    Length"
32170 QLINE=PEEK(QADDR)+256*PEEK(QADDR+1)
32180 IF QLINE>=32000 THEN END
32190 QLEN=0:QSUM=QLEN:QCNT=QLEN:? #QF;"    ";QLINE,"- ";
32200 IF NOT (QCNT<12 AND QLEN<500 AND QLINE<32000) THEN 32270
32220 QLEN=QLEN+PEEK(QADDR+2):QCNT=QCNT+1
32230 IF PEEK(QADDR+4)=0 AND QREM THEN QADDR=QADDR+PEEK(QADDR+2):GOTO 32260
32240 FOR QADDR=QADDR TO QADDR+PEEK(QADDR+2)-1
32250 QSUM=QSUM+PEEK(QADDR):NEXT QADDR
32260 Q$=STR$(QLINE):QLINE=PEEK(QADDR)+256*PEEK(QADDR+1):GOTO 32200
32270 QSUM=QSUM-676*INT(QSUM/676):QCNT=INT(QSUM/26)
32280 ? #QF;Q$,CHR$(65+QCNT);CHR$(65+QSUM-26*QCNT);"     ";QLEN
32290 GOTO 32180
```

TYPO TABLE

```
Variable checksum = 50796
    Line num range      Code    Length
    32000 - 32200        QD      518
    32220 - 32290        WQ      310
```

ATARI AT WAR

The wild world of conflict simulation

by KARL WIEGERS

It's no secret that the microcomputer has revolutionized the games people play. Game designers moved quickly from the prototypical **Pong** to the enormous selection of computer games available today, and as a result the TV screen has replaced many game boards. Most of the computer games that are currently available are arcade-style maze or shoot-'em-up games, but an increasing number of "adventure" games are being introduced. I use this term to include science fiction, fantasy and conflict-simulation games (the polite term for war games). War games have been used for generations by armies to simulate impending battles and to help formulate strategies and tactics.

WHAT IS A WAR GAME?

The war gamer has several hundred board games from which to choose, and they cover almost every conceivable conflict, both real and hypothetical, from ancient times to the far future. These games consist of four main features: a map that represents the battle area, cardboard counters or miniature figures that represent the opposing forces, dice and combat-results tables that are used to determine the outcome of battles, and a hefty book of rules. These games can be classified according to several schemes: the scale of combat, the historical period involved, or the type of combat simulated (naval, armored, air, etc.)

Karl E. Wiegers, Ph.D., is a research chemist for Eastman Kodak and an Atari hobbyist. He writes for a number of computer publications.

GAME-DESIGN CONSIDERATIONS

A war-game designer faces a continual struggle between the requirements of realism and playability. For example, a detailed board game of tactical air-to-air combat can involve many minutes of planning and execution by the players, even though a single game-turn might represent only a few seconds of real time. On the other hand, an Atari owner can take joystick in hand and play a real-time simulation of World War II fighter combat. The graphics and immediate response that a computer can provide offer a much better three-dimensional "feel" than is ever possible with a board game,

Air combat games for the Atari include **Spitfire Ace, Wingman,** and **Mig Alley Ace** (all MP). The latter two feature split screens in a two-player mode. The biplanes of World War I reappear in **Blue Max** (S), which features 3-D diagonal scrolling, **Flying Ace** (MG), and **Eagles** (SSI). And these games will soon be joined by **Fighter Command** (SSI), a very advanced reenactment of the Battle of Britain.

Similar features are included in games involving submarine combat, such as **Sea Dragon** (AI) and **Submarine Commander** (T). Sea Dragon scrolls through 24 sequential map screens to provide a vivid representation of undersea warfare. Submarine Commander allows you to evade depth charges and pursue enemy merchant ships.

Computerized combat games are more realistic than board games for other reasons as well. Combat units in most war games have a number of basic-strength values that represent capabilities for attack, defense, and overall morale. But board games usually don't account for smaller variations in the performance of a given unit under different sets of circumstances. In contrast, when you play **The Battle of Shiloh** (SSI) on your Atari, you can assign one of four different levels of risk (for both attack and defense) to each of your units. This greatly broadens the range of potential outcomes in each combat action.

A computer is also an ideal tool for introducing the "fog of war" into a war game. This term refers to the element of uncertainty and randomness that is a constant aspect of real combat and that adds spice and realism to a simulation. In **North Atlantic Convoy Raider** (MG) and **Midway Campaign** (MG), the computer controls enemy naval units which you must seek out and destroy while avoiding your own destruction. The computer limits the amount of informa-

continued on page 46

ILLUSTRATION: BILL BOYER


Boyer

continued from page 44

tion that you're able to obtain about its moves. This provides a high level of suspense, as well as an element of variation from game to game. In contrast, the randomizing techniques used in board games are relatively primitive: You roll dice to determine search results, invert units on the game board to mask their identities (but not their presence) from your opponent, and sometimes enlist a third person to serve as a referee while you and your antagonist sit in separate rooms.

A computer war-game designer must also take into account the concept of play balance; in other words, each player must have a chance to win the game, or it will soon lose its appeal. As Chris Crawford (the guru of game design at Atari) points out, the illusion of winnability must be present.

The designers of computer war games have solved this problem by incorporating randomizing techniques into the games and by providing for complex performance capabilities on the part of tactical units. These elements make for good balance in computer-war simulations. In the world of computer war games, the results at Waterloo or Gettysburg are not inevitable; there is still hope for the underdog. In addition, the different levels of difficulty offered by such games allow players of different abilities to introduce handicaps, and thus to maintain play balance.

THE PRO'S AND CONS OF THE COMPUTER

Most board games use a combat-results table (CRT) to resolve the vagaries of combat. The CRT shows the probabilities of various outcomes when a certain attack strength is matched against a particular defense under given conditions. A roll of the dice then selects one of these possible results. But a computer can provide a much more detailed tabulation of combat results than is possible with the tables used in board games.

In addition, the computational speed of the computer eliminates a great deal of the tedium from the unfolding of a war game. In **Eastern Front 1941** (APX), the Atari determines the results of many individual battles in a matter of seconds. Doing the same calculations by hand could easily involve thirty minutes of examining unit strengths, cross-referencing numbers in tables, and rolling dice. Even with the aid of the computer, war games tend to take much longer to play than either the designers or the players expected. However, use of the computer allows players to store their game and take a temporary leave of absence from the front.

In large board games, it often becomes tedious and time-consuming to monitor the status of your combat units. But many of the computer games mentioned here can instantly provide you with the tables that indicate the current status of your forces. The computer also eliminates the shuffling of paper that often plagues board games. In addition, the computer can monitor the supply status of each unit, as in games such as **Tigers in the Snow** (SSI). This adds dramatically to the realism and complexity of the simulation with virtually no sacrifice in playability.

One area in which board games are still far superior to computer war games is the map. Even the most outstanding applications of Atari graphics are hard-pressed to match the detailed and often beautiful maps used in many board games. Most of the maps used in board games feature a hexagonal grid that is superimposed upon the terrain to regulate movement and combat. Although, in principle, the computer can free you from such arbitrary restrictions, in fact such freedom is not yet found in many games. Even the sophisticated Eastern Front 1941 reverts to a mere modification of the primitive (by board game standards) use of squares to regulate the positioning of units. Unfortunately, there is only so much detail that can be placed into 48K of memory.

With a board game, it is possible to see all portions of a map (even a large map) at once, and to easily assess the overall situation. This is not possible with the small display area and the resolution offered by a TV screen. One way that game designers alleviate this problem is to create maps that can be scrolled, so that the TV screen is simply a window into a much larger playing area. Superb examples of map scrolling are found in Eastern Front 1941 and **Legionnaire** (MG), which matches ten of Caesar's fearsome legions against a computer-controlled army of barbarians.

Another solution is to provide a map and counters along with the computer game, as is done in **Tanktics** (MG) and **Dnieper River Line** (MG), two armored-combat games set in World War II.

Computer graphics simply cannot compete with paper and artwork when it comes to stationary displays. On the other hand, computer game maps can be altered simply by switching to a new character set of map symbols. In this way, Eastern Front 1941 shows the effects of the changing seasons with vivid visual displays that cannot be duplicated by the hardcopy maps of board games.

PICK A WAR, ANY WAR

Certain subjects are perennial favorites with war gamers, and the selection of games for the Atari reflects that fact. The Russian front in World War II, for example, holds a particular fascination, and games for the Atari are available on several levels: the strategic (Eastern Front 1941), the operational (Dnieper River Line), and the tactical (Tanktics). Another 1941-Russian-front simulation, **Panzers East** (MG), is due to be released soon.

Armored combat is always popular, as is witnessed by Tanktics, the upcoming tactical tank-to-tank World War II game **T.A.C. (MG), and Armor Assult** (E), which covers hypothetical tank combat in the near future. **Close Assult** (MG) simulates infantry combat in World War II on the tactical level. Another game on this subject, **Computer Ambush** (SSI), is scheduled for release soon. In this tactical-level game, you'll be able to equip soldiers with weapons and combat charcteristics of your own choosing.

A number of other significant World War II battles have captured the interest of computer gamers. These include the North African encounters of Rommel and Montgomery (**Knights of the Dessert**-SSI), the Battle of the Bulge (Tigers in the Snow), and the gigantic D-Day invasion (**Battle for Normandy**-SSI).

Dozens of board games can be found on the American Civil War, but only the Battle of Shiloh appears to have been

covered so far for the Atari. Both The Battle of Shiloh and **Shiloh 1862** (DY) deal with this critical battle in Tennessee. In Shiloh 1862, you direct Union soldiers and artillery against Confederate forces under General Atari.

An assortment of other wars from mankind's past and future can be refought or foreshadowed on your Atari. Legionnaire, for example, brings the campaigns of Caesar to your home computer. The Napoleonic wars are always popular; among the offerings in this genre are **Paris in Danger** (MG), **Leipzig 1813** (DY) and **Waterloo 1815** (DY). The upcoming **Broadsides** (SSI) is a game of naval combat from the Age of Sail; it features both arcade-style action and authentic simulations that allow you to make the kind of decisions that real sea captains once made. The Vietnam War is reenacted in **V.C.** (MG). You command air cavalry and field artillery units in a tactical simulation of the controversial war that so intimately engaged friend and foe, soldier and civilian.

ERA	TITLE	PUB.	SCALE	TYPE
Ancient	Legionnaire	MG	Operational	Land
Napoleonic	Paris in Danger	MG	Strategic/Tactical	Land
Napoleonic	Leipzig 1813	DY	Operational	Land
Napoleonic	Waterloo 1815	DY	Operational	Land
18th Cent.	Broadsides	SSI	Tactical	Sea
Civil War	The Battle of Shiloh	SSI	Operational	Land
Civil War	Shiloh 1862	DY	Operational	Land
World War I	Blue Max	S	Tactical	Air
World War I	Flying Ace	MG	Tactical	Air
World War I	Eagles	SSI	Tactical	Air
World War II	Hellcat Ace	MP	Tactical	Air
World War II	Spitfire Ace	MP	Tactical	Air
World War II	Fighter Commander	SSI	Strategic	Air
World War II	Submarine Commander	T	Tactical	Sea
World War II	Fathom's 40	DS	Tactical	Sea
World War II	North Atlantic Convoy Raider	MG	Operational	Sea
World War II	Midway Campaign	MG	Operational	Sea
World War II	Tanktics	MG	Tactical	Armor
World War II	Dneiper River Line	MG	Operational	Armor
World War II	T.A.C.	MG	Tactical	Armor
World War II	Eastern Front 1941	APX	Strategic	Land
World War II	Panzers East	MG	Strategic	Land
World War II	Knights of the Desert	SSI	Operational	Land
World War II	Tigers in the Snow	SSI	Operational	Land
World War II	Battle for Normandy	SSI	Operational	Land
World War II	Close Assault	MG	Tactical	Land
World War II	Computer Ambush	SSI	Tactical	Land
World War II	Operation Whirlwind	B	Tactical	Land
Korean War	MIG Alley Ace	MP	Tactical	Air
Vietnam War	V.C.	MG	Tactical	Land
Modern	Wingman	MP	Tactical	Air
Hypothetical	B-1 Nuclear Bomber	MG	Strategic	Air
Hypothetical	COMMBAT	AI	Tactical	Armor
Hypothetical	Sea Dragon	AI	Tactical	Sea
Hypothetical	Armor Assault	E	Tactical	Armor
Hypothetical	War	AI	Operational	Air/Land
Hypothetical	Nukewar	MG	Strategic	All
Hypothetical	NATO Commander	MP	Strategic	Land
Hypothetical	Germany 1985	SSI	Strategic	Land

Game publishers shown by their initials are:

AI = Adventure International
APX = Atari Program Exchange
B = Broderbund Software
DS = Datasoft
DY = Dynacomp
E = Epyx
MG = Microcomputer Games
MP = MicroProse Software
S = Synapse Software
SSI = Strategic Simulations, Inc.
T = Thorn E.M.I.

HYPOTHETICAL WAR

In addition to simulations of real wars, there are always hypothetical scenarios to consider, as if there hasn't been enough real conflict in human history to simulate. You're able to anticipate the outcome of a collision of the superpowers in **Germany 1985** (SSI). Similarly, **NATO Commander** (MP) is a real-time strategic simulation of a hypothetical Warsaw Pact invasion of Western Europe.

I mentioned earlier that the category of adventure games also includes science fiction and fantasy games. Several near-future hypothetical scenarios, such as the strategic-level game **Nukewar** (MG), bridge the gap between the recreation of past conflicts and the realm of pure conjecture.

CONCLUSION

As you can see, adapting detailed conflict-simulation games to a microcomputer is not easy. Many problems involving realism and playability must be resolved. Even so, a widening variety of these games is available for the Atari. The computer can greatly increase ease of play in such games by handling monotonous details, but even Atari graphics pale in comparison with those of a well-executed board game. Real-time simulations and 3-D games, especially those involving air and sea combat, are often much more fun on the computer, though. In fact, I think I'll go shot down some Zeroes now! Where's my flack jacket?


TAXAN
CORVUS SYSTEMS
ATARI 800

FAST FINGERS

Autoboot with a ghost at the keyboard

by CRAIG CHAMBERLAIN

Picture the following. Seated at your Atari computer, you insert the BASIC cartridge and turn the system on. The normal sequence of events ensues, whereupon the familiar prompt "READY" appears on the screen. Suddenly, you hear the keyboard clicking, as if invisible fingers were pressing the keys, and the words "OR NOT, HERE I COME!" appear. More clicking follows, and you see direct-mode commands to BASIC being typed in and immediately executed. You may even find that your disk drive or cassette machine has been activated to LOAD or RUN files! Is it a ghost? No, it's just Fast Fingers (FF), a revolutionary new concept in autoboot programs for the Atari.

ABOUT AUTOBOOT PROGRAMS

An autoboot program is one that is automatically executed when the computer is powered on. If you use DOS 2.0S in a disk-based system, you're probably aware that DOS looks for a program named AUTORUN.SYS upon booting. If AUTORUN.SYS exists on the disk, DOS will attempt to load and run it as a machine-language program before transferring control to a cartridge or loading DUP.SYS. If your system is cassette-based, you may have run autoboot programs by pressing and holding [START] as you turn on the computer.

The most common type of AUTORUN.SYS program, available as a one-sector file on the DOS 2.0S Master Disk, loads the RS-232 handler from the 850 interface if it's turned on. Another common type executes a line of BASIC commands by taking temporary control of the screen editor. Fast Fingers works by taking control of the keyboard instead, and lets you preprogram an entire sequence of keyboard entries (including cursor and screen editor controls and the [RETURN] key) to be executed as the autoboot file. The autoboot file can be executed with any language, including PILOT and the Assembler Editor cartridge, or even in MEMO PAD mode!

HOW TO USE FAST FINGERS

Type in the Fast Fingers listing below very carefully. Use TYPO to check for typing accuracy, and SAVE a copy and put it away for safekeeping before you RUN the program.

Now RUN it. The first prompt requests that you enter an input-device specification. For now, enter K for keyboard (a colon isn't necessary since there is no filename in this case). Next, the screen clears and the READY prompt appears. Don't be fooled — the program is still running. At this point, every keystroke that produces a legal ATASCII character is being recorded.

If you are using cassette, type CLOAD [RETURN] [RETURN] RUN [RETURN]. If you're using disk, type RUN "D:GAME [RETURN]. (Don't type a period after [RETURN], though.) You may replace GAME with the name of any BASIC tokenized file you wish to run.

Now both cassette and disk users should hold down [CTRL] and press [3]. This generates an end-of-file code, and terminates your input. A message is now printed that tells you how many keystrokes were recorded.

The next prompt asks whether you're using cassette or disk. Respond with C or D. If you're using disk, make sure the disk to which AUTORUN.SYS is to be written is in the drive. Next, both cassette and disk users should press [RETURN].

One of FF's special features is that it lets disk users append their AUTORUN.SYS file to one that already exists (e.g., the standard type that loads the RS-232 handler). If an AUTORUN.SYS file is present on the disk when you press [RETURN] (after the most-recently-described prompt), you'll see the prompt "APPEND?" Responding to this with Y appends your file to the end of the previously existing one. A possible application for this is loading the RS-232 handler and then automatically loading and running a terminal program. You could even have FF set your terminal configuration and dial a number for you!

The next two prompts ask for the load and patch addresses. Press [RETURN] for now to use the default parameters. These will be discussed more fully later in this article.

The last prompt asks for a speed value between one and nine. The fastest speed, One, means that keystrokes will be typed in at a maximum rate of sixty per second at boot time. Speed Nine types one key

continued on next page

PHOTO: LINDA TAPSCOTT AIRBRUSH: KYLE BOGERTMAN

every four seconds. Press [RETURN] for the default speed of Three.

After you press [RETURN] for the speed prompt, the program will write the boot file. Cassette users should make sure that you've depressed the RECORD and PLAY buttons on your recorder; when the computer beeps twice, press [RETURN]. When this is finished, the "real" READY prompt appears. At this point, cassette users should CSAVE a BASIC program without changing the tape's position. Disk users should make sure that your disk contains the game program and DOS files.

BOOTING – CASSETTE USERS

Turn everything off, including the 850 interface. Insert the cartridge into the computer. Then insert the cassette, rewind it and press the PLAY button. Next, hold down [START] and turn on the computer simultaneously. You can release the key when you hear the beep. Now press [RETURN]. The autoboot program will load and execute, whereupon the following sequence should occur (if you followed the example and typed in CLOAD and RUN previously):

1. The command CLOAD will appear on the screen as if it were being typed in on the keyboard. Accompanying keyclicks will sound, and [RETURN] will be pressed.

2. The computer will beep, [RETURN] will be pressed again and the BASIC program will load.

3. After the program is loaded, the command RUN will appear, [RETURN] will be pressed and the program will start.

BOOTING – DISK USERS

The disk booting procedure is quite simple. DOS, when loaded, loads and executes AUTORUN.SYS, which takes over the keyboard and runs your program with the command previously given, RUN"D:GAME.

APPLICATIONS

One interesting experiment is to have Fast Fingers type in a program for you, then RUN it and respond to prompts. When in record mode, type in a short program that requests user input, then type RUN, and then type a response for the programmed input. Note that, when in record mode, immediate-mode BASIC commands do not take effect. While you're typing in the program, try using cursor controls to change a line after it's been entered. When the autoboot program executes, you'll see your program being entered, edited, RUN, and responded to, as if by a ghost!

If you've written a menu-oriented program that requires keyboard input, FF provides an ideal auto-demonstration mode for showing someone how the program works. Fast Fingers can provide an entire sequence of keystrokes that demonstrate the program's various features, and the original program need not be modified at all.

Librarians for user's groups will find this next application to be very useful. By creating an autoboot file that consists of pairs of commands, such as LOAD "D:filename [RETURN] and CSAVE [RETURN] [RETURN], you can transfer groups of disk files to cassette automatically, while leaving the computer unattended!

ALTERNATE INPUT DEVICE

At the first prompt (for input-device specification), you may supply the name of an alternate input device, such as C or D:filename. This allows you to prepare and save a text file for use with Fast Fingers beforehand. If you supply an alternate specification, Fast Fingers will load the file and display it on the screen.

LOAD AND PATCH ADDRESSES

The default load address for cassette systems is 1792 (decimal); for disk systems it is 13312. The initialization code that installs the patch (57 bytes), the patch itself (56 bytes), and the keystroke data (one byte per keystroke) are loaded at this address. Thus, the number of contiguous free bytes needed to load the program is 113 plus the keystroke count. If, for example, you're using cassette and your application requires the RS-232 handler, which loads at 1792 (decimal), you must specify a higher load address such as 4000 (decimal).

The "patch address" is the address at which the FF machine-code routine (the patch) and the keystroke data are stored. The default value is 1536 (decimal), which is the start of Page 6, an area of memory that is usually free for programmer's user. If your program uses Page 6, or your keystroke count exceeds 200, you must locate a patch elsewhere. One possible location would be above a lowered RAMTOP.

HOW IT WORKS

Every sixtieth of a second (or one "jiffy"), after the monitor has finished drawing one frame, the Atari Operating System (OS) generates a Vertical Blank Interrupt. This period of time is very short by human standards, but it's long enough for the computer to take care of a number of important housekeeping operations. One such task is to check the keyboard for a keypress. Whenever a key is pressed, the corresponding hardware keycode of that key is stored in location 764 (decimal). When the OS is asked to input a character from the keyboard, it gets the keycode from location 764, converts it to an ATASCII character, and sets location 764 to 255, which means that the key has been read. FF also monitors location 764 during the vertical blank. If it finds a 255, it gets the keycode for the next character from its data buffer and places it in location 764. This makes the OS think that a key has been pressed. FF continues to do this until the data buffer is empty, whereupon control is passed to BASIC or the current application.

LIMITATIONS OF FAST FINGERS

FF will work only with legal ATASCII characters. Therefore, the following keystrokes should not be used: [CTRL]-[1], [CTRL]-[3] through [CTRL]-[0], [CTRL]-[?], [CAPS/LOWR], the inverse key, all keys with both [SHIFT] and [CTRL] pressed, and the [SHIFT] and [CTRL] keys pressed alone.

FF supports lower case by automatically generating a keycode for the [CAPS/LOWR] key the first time a lowercase letter is used. It will remain in the lower-case mode until it reaches the end of the keycode data. Upper case will still work, though, because the keycodes imply that [SHIFT] is being pressed.

Inverse video is also supported, but it

creates a minor problem. When input data alternates between inverse and normal characters, FF generates a keycode for the inverse key. But if the switch between normal and inverse video occurs at the beginning of a line in BASIC, the character after this keycode occasionally goes undetected by the system at boot time (when higher speeds for keypresses are used). I haven't been able to discover the reason for this. Possible solutions are to press the inverse key before pressing [RETURN] on the previous line, to make the first inverse character optional (such as a space), or to use a slower speed.

FF can call the MEMO PAD (the command is BYE), but cannot return to BASIC by itself. It can also call DUP (the command is DOS), but it will be deactivated when it returns to BASIC.

When entering text from the keyboard in Fast Fingers, you need press [ESCAPE] only once to invoke that character, rather than two times as is usually the case. To make FF produce only one [ESC] keystroke, remove the following two statements from line 310 of the program:

B$(L) = CHR$(28):L = L + 1

If you want to use the end-of-file keycode ([CTRL]-[3]) in your data file, change the Q = 160 in line 350 to Q = 154 and press [CTRL]-[,] when the EOF keycode is desired.

FF cannot simulate a press of the [BREAK] key.

While the autoboot program created by FF is running, the keyboard is disabled to prevent a conflict with the keycodes coming from FF. You can change this with POKEs to locations 16 (POKMSK) and 53774 (IRQEN). POKE both with 64 to enable the keys, 128 to enable [BREAK], 192 to enable both, and 0 to disable both. The keyboard is automatically re-enabled when FF runs out of keystroke data.

SUMMARY

Your imagination is the only limit to Fast Finger's possibilities. Given FF's practically automatic implementation, it's an easy matter to make your computer execute any command sequence, run programs, or simply perform acrobatics with the cursor. Turning on your computer was never so much fun!

Requires 16K RAM

```
95 REM ***** FAST FINGERS *****
96 REM * BY CRAIG CHAMBERLAIN *
97 REM *    ANTIC MAGAZINE     *
98 REM *    FEBUARY, 1984      *
99 REM ************************
100 ? CHR$(125):? "FAST FINGERS BOOT F
ILE MAKER":? "by Craig Chamberlain 8/1
/83":?
110 DIM M(112),A(70),F$(16):FOR K=0 TO
 112:READ P:M(K)=P:NEXT K:FOR K=1 TO 7
0:READ P:A(K)=P:NEXT K
120 DIM B$(FRE(0)-250):L=1:SHIFT=64:CT
RL=128:MODE=0:LC=0
200 ? "PLEASE ENTER INPUT DEVICE SPEC"
:INPUT F$:IF F$="" THEN 200
210 ? CHR$(125):? "READY":TRAP 700:OPE
N #1,4,0,F$:TRAP 460
300 GET #1,P:? CHR$(P);
310 IF P=27 THEN B$(L)=CHR$(28):L=L+1:
Q=28:GOTO 450
315 IF P>27 AND P<32 THEN Q=A(P+28):GO
TO 450
320 IF P>124 AND P<128 THEN Q=A(P-60):
GOTO 450
325 IF P>154 AND P<160 THEN Q=A(P-95):
GOTO 450
330 IF P>252 THEN Q=A(P-185):GOTO 450
340 INV=0:IF P>127 THEN P=P-128:INV=1
345 IF INV<>MODE THEN MODE=1-MODE:B$(L
)=CHR$(39):L=L+1
350 IF P=0 THEN Q=160:GOTO 450
355 IF P<27 THEN Q=CTRL+A(P):GOTO 450
360 IF P=32 THEN Q=33:GOTO 450
365 IF P<40 THEN Q=SHIFT+A(P-5):GOTO 4
50
370 IF P=40 THEN Q=SHIFT+A(36):GOTO 45
0
375 IF P=41 THEN Q=SHIFT+A(27):GOTO 45
0
380 IF P<48 THEN Q=A(P-5):GOTO 450
385 IF P<58 THEN Q=A(P-21):GOTO 450
390 IF P<65 THEN Q=A(P-15):GOTO 450
395 IF P<91 THEN Q=SHIFT+A(P-64):GOTO
450
400 IF P<97 THEN Q=A(P-41):GOTO 450
405 IF P=123 THEN Q=130:GOTO 450
410 IF P=124 THEN Q=79:GOTO 450
415 IF LC=0 THEN LC=1:B$(L)=CHR$(60):L
=L+1
420 Q=A(P-96)
450 B$(L)=CHR$(Q):L=L+1:GOTO 300
460 IF PEEK(195)<>136 THEN 700
470 B$(L)=CHR$(255):CLOSE #1:POKE 694,
0:POKE 702,64:?
480 ? "TOTAL: ";L;" KEYSTROKES":IF L>2
00 THEN ? "WARNING: TOO BIG FOR PAGE S
IX"
500 ? :? "CASSETTE OR DISK";:INPUT F$:
P=ASC(F$):IF P<67 OR P>68 OR F$="" THE
N 500
510 C=11*(P=67):LA=13312:IF C THEN LA=
1792:GOTO 520
512 WM=8:OPEN #1,6,0,"D:AUTORUN.SYS":I
NPUT #1;F$:CLOSE #1:IF F$(2,2)<>" " TH
EN 520
514 ? "APPEND";:INPUT F$:IF F$="" THEN
 ? "NO":GOTO 520
516 WM=8+(F$(1,1)="Y"):IF F$(1,1)<>"Y"
 AND F$(1,1)<>"N" THEN 514
520 ? "LOAD ADDRESS (";LA;") ";:TRAP 5
30:INPUT LA
530 PA=1536:? "PATCH ADDRESS (";PA;")
";:TRAP 540:INPUT PA
540 I=3:? "SPEED 1-9 (";I;") ";:TRAP 5
50:INPUT I:IF I<1 OR I>9 OR I<>INT(I)
THEN 540
550 P=1:FOR K=1 TO I:P=P*2:NEXT K:M(60
)=P/2-1
```

continued on next page

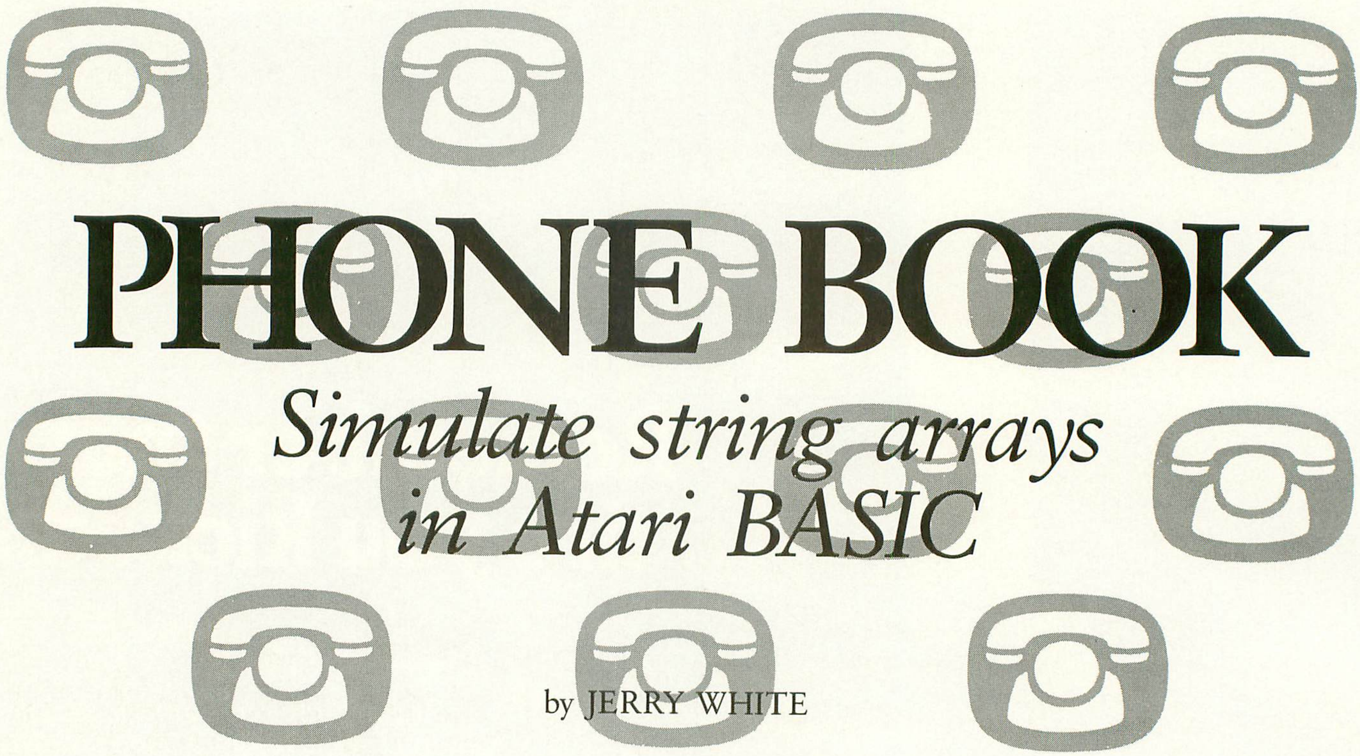
```
560 ? :LA=LA+C:A=LA+57:GOSUB 710:A=PA:GOSUB 710:A=LA+113:GOSUB 710:A=PA+56:GOSUB 710:A=LA+29:GOSUB 710:A=LA+32
570 GOSUB 710:A=PA+56:GOSUB 710:A=PA+13:GOSUB 710:A=PA+14:GOSUB 710:M(49)=INT(PA/256):M(51)=PA-256*INT(PA/256)
600 LA=LA-C:LO=LA-256*INT(LA/256):HI=INT(LA/256):TRAP 700:IF C THEN 630
610 OPEN #1,WM,0,"D:AUTORUN.SYS":PUT #1,255:PUT #1,255:PUT #1,LO:PUT #1,HI
620 EA=LA+112+L:PUT #1,EA-256*INT(EA/256):PUT #1,INT(EA/256):GOTO 650
630 OPEN #1,8,128,"C:":PUT #1,0:PUT #1,INT((L+250)/128):PUT #1,LO:PUT #1,HI
640 PUT #1,PA+55-256*INT((PA+55)/256):PUT #1,INT((PA+55)/256):FOR K=1 TO 5:READ P:PUT #1,P:NEXT K
650 FOR K=0 TO 112:PUT #1,M(K):NEXT K:? #1;B$;:IF C THEN 690
660 PUT #1,226:PUT #1,2:PUT #1,227:PUT #1,2:PUT #1,LO:PUT #1,HI
690 CLOSE #1:TRAP 40000:CLR :END
700 ? "ERROR ";PEEK(195):TRAP 40000:CLR :END
710 READ I:M(I)=A-256*INT(A/256):M(I+1)=INT(A/256):RETURN
800 DATA 165,16,41,63,133,16,141,14,210,162,55,189,57,6,157,57,6,202,16,247,142,252,2,232,142,255,2,189,113,6,157
801 DATA 113,6,232,208,6,238,29,6,238,32,6,201,255,208,237,169,7,162,6,160,57,32,92,228,24,96,165,20,41,1,208,46
802 DATA 174,252,2,232,208,40,174,113,6,238,70,6,208,3,238,71,6,142,252,2,232,208,23,169,64,141,190,2,165,16,9
803 DATA 192,133,16,141,14,210,169,7,162,228,160,98,32,92,228,76,98,228,96
810 DATA 63,21,18,58,42,56,61,57,13,1,5,0,37,35,8,10,47,40,62,45,11,16,46,22,43,23,50,31,30,26,24,29,27,51,53
811 DATA 48,7,6,32,14,34,38,66,2,54,15,55,102,117,96,70,98,71,78,162,142,143,134,135,12,116,119,172,108,118,52
812 DATA 44,158,180,183
820 DATA 12,15,28,31,37,40,70,73,78
830 DATA 169,60,141,2,211
```

TYPO TABLE

Variable checksum = 342414

Line num range	Code	Length
95 - 210	LA	565
300 - 360	OH	509
365 - 420	YT	493
450 - 514	ZQ	518
516 - 560	PO	505
570 - 630	GW	529
640 - 800	XF	553
801 - 811	GK	500
812 - 830	GV	74


Remember that tire you kicked...?
It certainly remembers, and it's not exactly happy about it. And your steam iron? It's sick of being tied up. The way you wrap that cord tightly around it and then shove it in the closet. They're mad. Fighting mad. And they're organizing for the attack in Megamania for Atari home computers and the 5200.
Faster and faster they come, and from everchanging directions. You've got to shoot to stay alive.
They've organized those ice cream sandwiches you left in the back of the freezer, the dice you said were bad luck.
Your peaceful home is a hotbed of discontent. And telling the tire you're sorry isn't going to help.
Designed by Steve Cartwright, adapted by Glyn Anderson.
Megamania™ for Atari® Home Computers and the 5200.™
Atari® and 5200™ are trademarks of Atari, Inc. ©1983 Activision, Inc.
ENERGY
16248
MEGAMANIA
ACTIVISION
We put you in the game.



PHONE BOOK

Simulate string arrays in Atari BASIC

by JERRY WHITE

Many ANTIC readers have asked how to simulate a string array in Atari BASIC. This program, called TELEPHON, will serve as a tutorial on string arrays and string manipulation. But for those of you who just want to type in and use this handy little program, I'll begin with some instructions.

PROGRAM INSTRUCTIONS

TELEPHON will run on any Atari computer with BASIC installed. Either a cassette recorder or disk drive is required. A printer is optional.

The program begins by asking where your data will be stored. Type [C] for cassette or [D] for disk. TELEPHON uses INPUT commands and does not have a sophisticated keyboard handler. Be sure to respond using only normal (not inverse), uppercase characters, and press the [RETURN] key after each entry.

A menu of choices will appear on the screen. A display near the top of the screen shows the number of records currently in memory and the number of records available. A record consists of a name and a telephone number. The information in each record is broken up into five "fields," as shown in Table 1. Also shown are the corresponding variable names and lengths for each field.

Table 1

FIELD	STRING	MAXLEN
Last name	LAST$	12
First name	FIRST$	12
Area code	AREA$	3
Exchange	EXCH$	3
Number	PNUM$	4

We do not yet have a data file, so we begin with Menu Choice 1, which directs the program to enter new data. This is also the choice to use whenever you add records. After you have created and saved a TELEPHON data file by writing it onto tape or disk, you will begin each session by using Menu Choice 2 to load your data into memory.

You will then be prompted to enter data into the fields described above. After the five items have been entered, the program will return to the menu screen. Notice that the display near the top of the screen now indicates that one record is in memory and that one fewer record is available. Enter a few more records using Menu Choice 1, but not too many. We want to make sure that your program is working perfectly before you type in your whole telephone book.

After entering a few records, save your data by using Choice 3 on the menu. A message will appear on the screen that tells you to press the [START] key when your input/output device is ready. Cassette users should insert a blank tape, rewind it to the beginning, and then press the [PLAY] and [RECORD] buttons simultaneously. Disk users must make sure that a diskette with an ample number of free sectors is ready in Drive 1. Our program will use the disk filename "TELEPHON.DAT", but you can choose your own filename. When the [START] key is pressed, cassette users will hear the familiar series of beeps. Press [RETURN] to begin the process of writing your data onto tape.

Whenever you load data using Menu Choice 2, any records in memory will be lost. Be sure that you always use this choice first.

If you make an error while entering data, Choice 4 allows you to correct it.

Jerry White is a Contributing Editor to ANTIC. His programs are available from several software companies, notably Adventure International and Educational Software.

The program will ask for the "last name" of the record to be altered, and will search through each record until a match is found. Upon finding a match, the program displays an entire record. If this is the record to be altered, respond with [Y]; if not, respond with [N]. If you type [N], the search will continue. If you type [Y], you may then reenter the data correctly. This procedure can also be used to update your files when your friends change their telephone numbers.

Menu Choice 5 tells the program to sort your data by last name. This makes it easier to find a specific record when you print your data using Menu Choice 6.

Those of you with printers will find this program particularly useful. You will be asked to type [S] to display your telephone directory on the screen, or [P] to include a listing to your printer. The listing can replace your old personal telephone book.

To freeze the display while using Choice 6, press the [CTRL] and [1] keys. Repeat this procedure to continue.

Menu Choice 7 ends the program and returns control to the BASIC cartridge.

TELEPHON was written entirely in Atari BASIC. The [SYSTEM RESET] and [BREAK] keys function normally when it is in use. If you accidentally press the [BREAK] key, you can return to the menu screen by entering the following command in immediate mode:

```
GOTO MENU
```

If you'd like to disable the [BREAK] key, add the following line to the program:

```
14001 POKE 16,64:POKE 53774,112
```

A WORD OF WARNING

Before we move on to the string array tutorial and some of the more technical information, there is one important point that you should understand. TELEPHON dimensions a string called ARRAY$ according to the current amount of free memory in your Atari. This will leave very little RAM available for additional lines of code. If you [BREAK] into the program to add code, you should first issue a CLR command in immediate mode. This will make all of the string area available again, but it will also erase any data you have in memory. Therefore, you should always SAVE your data, and the program itself, before making any changes to the program.

SIMULATING STRING ARRAYS

Atari BASIC is often derided by people who are familiar with Microsoft BASIC because it doesn't offer string arrays, although it does feature virtually unlimited string length and numeric arrays. BASIC XL, from Optimized Systems Software, offers both forms of string handling. But if you have only Atari BASIC, don't despair. String arrays *can* be simulated, and the TELEPHON program is a good example of this.

HOW TELEPHON WORKS

TELEPHON uses 34 characters for each record. The first 12 positions are assigned to the last name, and unused positions are padded with blanks. Positions 13 through 24 are used for the first name, 25 through 27 for the area code, 28 through 30 for the exchange, and 31 through 34 for the remaining four digits of the telephone number.

Our data is stored in the string called ARRAY$. ARRAY$(1,34) contains our first record, and is the equivalent of the first element in a Microsoft array. The equivalent of array element number two would be ARRAY$(35,68). To find the tenth element (record) in Atari BASIC, we can simply set a variable that is equal to our record length multiplied by 10, and then print the record using the following routine:

```
STRPOS = 10*34:
?ARRAY$(STRPOS,STRPOS + 33)
```

MEMORY-SAVING TECHNIQUES

Our data and the TELEPHON program must share the computer's memory, so I've kept the program as small as possible. If the program used all available RAM, there would be no room for data. As written, TELEPHON can store over 800 records of data on a 48K machine. If you delete the REM statements, you can salvage enough memory for about a dozen additional records.

One of the other memory-saving techniques I used in TELEPHON was to use the variable O instead of the number 0 and the variable I instead of the number 1. Six bytes of RAM are saved each time that a numeric variable replaces a number. These variables and their numeric values look similar, so they don't detract very much from program readability.

By studying this program carefully, you can learn how to manipulate strings in Atari BASIC. Table 2 provides a listing of previously undefined key variables and strings, along with brief definitions. This will help you understand the program's logic.

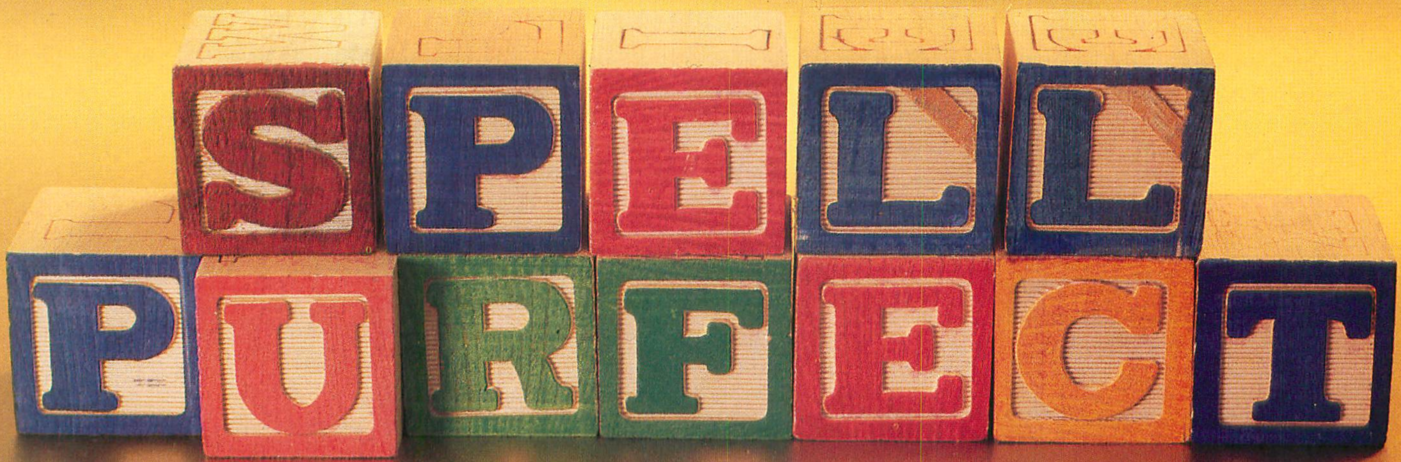
Do you have any other questions on Atari BASIC? Would you like to see other programs like this one in future issues of ANTIC? Please send your questions and comments to Jerry White in care of ANTIC Magazine.

Table 2

VAR/STR$	DEFINITION
ARRAY	Maximum number of records
BLANK$	Stores 34 spaces
CHOICE	Numeric value of chosen option
CHOICE$	String value of keyboard response
CLEAR	Clear-screen routine (line 14000)
DEV$	Stores C: or D: TELEPHON.DAT
DING	Sound routine (line 17000)
INLEN	Length of keyboard input
MENU	Menu routine (line 12030)
PAPER	Printer flag
REC$	Temporary record storage
RECLEN	Record length of 34
RLM1	Record length minus 1
STRPOS	String position
TABLEN	RAM location 201 (tab length)
TOTAL	Current total of records in memory
WORK$	Temporary storage used by sort

continued on page 57

Correction.



It's simple.

New **Spell Perfect**™ from LJK does what your second grade teacher wanted to do. It corrects your spelling. Used with Letter Perfect™, it checks your work against an expandable dictionary. Gives "sounds like" suggestions. Counts words. Add words. Technical terms. States and towns.

You can do it all, all-in-one dictionary.

When it comes to practical software for Ataris, Apples and look alikes, Spell Perfect is simple to learn. And hard to beat. Ask your dealer for a demonstration, or write LJK for more information.

LJK™

LJK, INC. • 7852 Big Bend Blvd., St. Louis, MO 63119 • (314) 962-1855

PHONE BOOK continued from page 55

Requires 16K RAM, 48K recommended

```
1 REM * TELEPHONE DIRECTORY *
2 REM *    BY JERRY WHITE    *
3 REM *    ANTIC MAGAZINE    *
4 REM *    FEBUARY, 1984     *
100 RECLEN=34:I=1:RLM1=RECLEN-I:GOTO 1
0030
1000 REM
1010 REM SORT ROUTINE
1020 REM
1030 GOSUB CLEAR:? :? "SORTING";:V=TOT
AL:IF V<2 THEN GOTO MENU
1040 V=INT(V/3)+I:FOR ME=I TO TOTAL-V:
? ".";:FOR L=ME TO I STEP -V:STRPOS=L*
RECLEN:LV=(L+V)*RECLEN
1050 IF ARRAY$(STRPOS-RLM1,STRPOS)<=AR
RAY$(LV-RLM1,LV) THEN POP :GOTO 1080
1060 WORK$=ARRAY$(STRPOS-RLM1,STRPOS):
ARRAY$(STRPOS-RLM1,STRPOS)=ARRAY$(LV-R
LM1,LV)
1070 ARRAY$(LV-RLM1,LV)=WORK$:NEXT L
1080 NEXT ME:IF V>I THEN 1040
1090 GOSUB DING:GOTO MENU
2000 REM
2010 REM READ DATA ROUTINE
2020 REM
2030 GOSUB CLEAR:? :? "READ DATA:":GOS
UB 16000
2040 CLOSE #I:OPEN #I,4,0,DEV$:TOTAL=0
2050 INPUT #I,REC$:TOTAL=TOTAL+I:? "RE
CORD #";TOTAL
2060 STRPOS=TOTAL*RECLEN-RLM1:ARRAY$(S
TRPOS,STRPOS+RLM1)=REC$:? REC$:? :GOTO
 2050
3000 REM
3010 REM WRITE DATA ROUTINE
3020 REM
3030 GOSUB CLEAR:IF  NOT TOTAL THEN ?
:? "I HAVE NO DATA TO WRITE":GOTO 1500
0
3040 ? :? "WRITE DATA:":GOSUB 16000
3050 CLOSE #I:OPEN #I,8,0,DEV$
3060 FOR ME=I TO TOTAL:? "RECORD #";ME
3070 STRPOS=ME*RECLEN-RLM1:REC$=ARRAY$
(STRPOS,STRPOS+RLM1)
3080 ? REC$:? #I;REC$:NEXT ME
3100 CLOSE #I:GOTO 15000
4000 REM
4010 REM ALTER DATA ROUTINE
4020 REM
4030 GOSUB CLEAR:? :? "ALTER DATA:"
4040 IF  NOT TOTAL THEN ? :? "I HAVE N
O DATA TO ALTER":GOTO 15000
4050 ? :? "ENTER LAST NAME FOR SEARCH"
;:INPUT LAST$
4060 INLEN=LEN(LAST$):IF  NOT INLEN TH
EN 4050
4070 ? :? "SEARCHING";
4080 FOR ME=I TO TOTAL:STRPOS=ME*RECLE
N-RLM1:? ".";
4090 IF LAST$=ARRAY$(STRPOS,STRPOS+INL
```

continued on next page

```
EN-I) THEN 4110
4100 NEXT ME:? :? :? LAST$;" NOT FOUND":GOTO 15000
4110 ? :? :? ARRAY$(STRPOS,STRPOS+RLM1)
4120 ? :? "IS THIS THE RECORD TO BE ALTERED (Y/N)";:INPUT CHOICE$
4130 IF CHOICE$="N" THEN 4100
4140 IF CHOICE$="Y" THEN ARRAY$(STRPOS,STRPOS+RLM1)=BLANK$:GOTO 6050
4150 GOTO 4110
5000 REM
5010 REM PRINT DATA ROUTINE
5020 REM
5030 GOSUB CLEAR:IF  NOT TOTAL THEN ?:? "I HAVE NO DATA TO PRINT":GOTO 15000
5040 ? :? "PRINT DATA:":PAPER=0
5050 ? :? "TYPE S FOR SCREEN DISPLAY ONLY"
5060 ? "TYPE P TO USE SCREEN AND PRINTER";:INPUT CHOICE$
5070 IF CHOICE$="S" THEN 5100
5080 IF CHOICE$="P" THEN PAPER=I:GOTO 5100
5090 GOTO 5050
5100 GOSUB CLEAR:? :LINE$(25,25)=" ":LINE$(29,29)="-":LINE$(33,33)="-"
5110 FOR ME=I TO TOTAL
5120 STRPOS=ME*RECLEN-RLM1:REC$=ARRAY$(STRPOS,STRPOS+RLM1)
5130 LINE$(I,24)=REC$(I,24)
5140 LINE$(26,28)=REC$(25,27)
5150 LINE$(30,32)=REC$(28,30)
5160 LINE$(34,37)=REC$(31,34)
5170 ? LINE$:IF PAPER THEN LPRINT LINE$
5180 NEXT ME
5200 GOTO 15000
6000 REM
6010 REM ENTER NEW DATA ROUTINE
6020 REM
6030 GOSUB CLEAR:TOTAL=TOTAL+I:STRPOS=TOTAL*RECLEN-RLM1
6035 IF TOTAL>ARRAY THEN TOTAL=TOTAL-I:? :? "NO ADDITIONAL RECORDS AVAILABLE":GOTO 15000
6040 ? :? "ENTER NEW DATA:"
6050 ? :? "LAST NAME:";:INPUT LAST$
6052 INLEN=LEN(LAST$):IF  NOT INLEN THEN 6050
6054 ARRAY$(STRPOS,STRPOS+INLEN-I)=LAST$
6060 ? :? "FIRST NAME:";:INPUT FIRST$
6062 INLEN=LEN(FIRST$):IF  NOT INLEN THEN 6060
6064 ARRAY$(STRPOS+12,STRPOS+INLEN+11)=FIRST$
6070 ? :? "AREA CODE:";:INPUT AREA$
6072 INLEN=LEN(AREA$):IF  NOT INLEN THEN 6070
6074 ARRAY$(STRPOS+24,STRPOS+INLEN+23)=AREA$
6080 ? :? "EXCHANGE:";:INPUT EXCH$
6082 INLEN=LEN(EXCH$):IF  NOT INLEN THEN 6080
6084 ARRAY$(STRPOS+27,STRPOS+INLEN+26)=EXCH$
6090 ? :? "NUMBER:";:INPUT PNUM$
6092 INLEN=LEN(PNUM$):IF  NOT INLEN THEN 6090
6094 ARRAY$(STRPOS+30,STRPOS+INLEN+29)=PNUM$
6100 GOSUB DING:GOTO MENU
7000 REM
7010 REM EXIT TO BASIC
7020 REM
7030 GOSUB CLEAR:? :? "BASIC":? "IS";:END
10000 REM
10010 REM PROGRAM INITIALIZATION
10020 REM
10030 DIM LAST$(12),FIRST$(12),AREA$(3),EXCH$(3),PNUM$(4),BLANK$(RECLEN)
10040 DIM REC$(RECLEN),WORK$(RECLEN),DEV$(14),CHOICE$(I),LINE$(38)
10050 ARRAY=INT(FRE(0)/RECLEN)-2:DIM ARRAY$(ARRAY*RECLEN)
10060 ARRAY$=" ":ARRAY$(ARRAY*RECLEN)=" ":ARRAY$(2)=ARRAY$:BLANK$=ARRAY$(I,RECLEN)
10070 CLEAR=14000:MENU=12030:LET TABLEN=201:DING=17000
11000 GOSUB CLEAR
11010 POKE 82,I:POKE 83,39
11020 ? :? "TYPE C FOR CASSETTE OR D FOR DISK";:INPUT CHOICE$
11030 IF CHOICE$="C" THEN DEV$="C:":GOTO 12000
11040 IF CHOICE$="D" THEN DEV$="D:TELEPHON.DAT":GOTO 12000
11050 GOTO 11020
12000 REM
12010 REM MENU ROUTINE
12020 REM
12030 GOSUB CLEAR:SETCOLOR 2,12,0:SETCOLOR 4,12,4
12040 POKE TABLEN,7
12050 POKE 752,I:? :? ,"ANTIC TELEPHONE DIRECTORY"
12060 POKE TABLEN,9
12070 ? :? ,"RECORDS IN MEMORY=";TOTAL
12080 ? ,"RECORDS AVAILABLE=";ARRAY-TOTAL
12090 POKE TABLEN,15
12100 ? :? ,"M E N U"
12110 POKE TABLEN,10
12120 ? :? ,"(1) ENTER NEW DATA"
12130 ? :? ,"(2) READ IN DATA"
12140 ? :? ,"(3) SAVE DATA"
12150 ? :? ,"(4) ALTER DATA"
12160 ? :? ,"(5) SORT DATA"
12170 ? :? ,"(6) PRINT DATA"
12180 ? :? ,"(7) EXIT TO BASIC"
12190 TRAP 12190:POSITION 10,22:? CHR$(156):CHOICE=0:POKE 752,0
12200 POKE 764,255:POSITION 10,22:GOSUB DING:? "ENTER YOUR CHOICE";:INPUT CH
```

```
OICE$
12210 CHOICE=VAL(CHOICE$):IF CHOICE<I
OR CHOICE>7 THEN 12190
12220 TRAP 20000:ON CHOICE GOTO 6030,2
030,3030,4030,1030,5030,7030
13000 REM
13010 REM MISC. SUBROUTINES
13020 REM
14000 GRAPHICS 0:SETCOLOR 2,CHOICE,0:S
ETCOLOR 4,CHOICE,4:SETCOLOR I,0,13
14010 RETURN
15000 ? :? "PRESS START FOR MENU":POKE
 53279,0:GOSUB DING
15010 IF PEEK(53279)<>6 THEN 15010
15020 GOTO MENU
16000 ? :? "PRESS START WHEN I/O DEVIC
E IS READY":POKE 53279,0:GOSUB DING
16010 IF PEEK(53279)<>6 THEN 16010
16020 GOSUB CLEAR:RETURN
17000 FOR VOLUME=15 TO 0 STEP -I:SOUND
 0,CHOICE,2,VOLUME:NEXT VOLUME
17010 RETURN
20000 IF PEEK(195)=136 THEN CLOSE #I:G
OTO 15000
20010 ? :? "ERROR ";PEEK(195);" AT LIN
E ";PEEK(186)+256*PEEK(187):GOTO 15000
```

TYPO TABLE

Variable checksum = 1438193

Line num range	Code	Length
1 - 1060	AT	378
1070 - 3010	QT	259
3020 - 4030	LZ	266
4040 - 4150	DG	371
5000 - 5110	JX	358
5120 - 6030	ZY	286
6035 - 6080	GN	353
6082 - 10010	XF	244
10020 - 11050	RR	422
12000 - 12110	MK	274
12120 - 13000	VF	436
13010 - 17010	RH	327
20000 - 20010	MV	119

CHARACTER GRAPHICS

Redefine the set anyway you choose

by CHRIS CHABRIS

The Operating System of every Atari computer comes with predefined "shapes" for up to 128 characters. These are normally displayed as white characters against a blue background. These characters include upper- and lower-case letters, numbers, punctuation, and graphics characters. These 128 characters may also be displayed in "inverse video" (a blue character against a white background), a feature which effectively doubles the number of discretely different characters available for screen display.

Some of you may not know, however, that you can customize this character set. This customizing feature, which is not provided by many other home computers, not only allows you to create different text characters, such as those found in foreign alphabets, it also allows you to create text objects for applications such as animated multicolor game shapes! Redefined characters are really quite easy to make and use in Atari BASIC.

First, a few definitions you will need to understand the techniques:

Character Sets: A character set is a table located in memory which defines the shapes of the characters to be displayed in Graphics Modes 0, 1, and 2. The character set supplied by Atari contains the patterns for 128 characters and resides permanently in the Operating System ROM cartridge.

Memory Pages: A "page" of memory consists of 256 contiguous bytes of memory whose starting address (the location of the first byte) is evenly divisible by 256. Page Zero, then, represents the 256 memory locations starting at location 0 in RAM and ending at location 255; Page Six starts at 1536 and ends at 1791.

Bits: A byte of memory is divided into eight "bits," each of which can be either zero or one. A bit equal to one is normally said to be "on." The positions of the bits in a byte are numbered 0 through 7 starting at the right and moving leftward. The decimal value of a byte, which may range from 0 to 255, is found by adding together the value of the "on" bits. An "on" bit in position 0 equals one, "on" in position 1 equals two, "on" in position 2 equals four, "on" in position 3 equals eight, etc. For example, eight bits of a byte may look like 01001101. Bits 6, 3, 2, and 0 are "on." Thus $2^6 + 2^3 + 2^2 + 2^0 = 77$, the decimal value of this byte. (Remember: two to the zero power equals one.)

BUILDING CHARACTERS

The easiest way to begin using character graphics is to follow a hands-on, step-by-step method for incorporating them into your programs. You will need some graph paper to plot out the new character shapes.

Step 1 — Decide what you want your new characters to look like. As an example, let's say we want to design a character for capital omega, the last letter of the Greek alphabet.

Step 2 — Draw your character on the graph paper. Each character occupies an eight-by-eight grid which represents eight rows of bytes (vertical) and eight columns of bits (7 through 0 horizontal). Each one of the 64 bits in this character square represents one pixel (short for "picture element") that can be turned on or off at the programmer's command. Keeping this in mind, let's design the Greek letter omega,

continued on page 62

Christopher Chabris corrects his previous bio. He has not yet graduated from high school, but does work as a Junior Programmer for CSIBM, a computer service company in Yonkers, N.Y. He is a Chess Master candidate, and has represented the U.S. in international matches.

LET US ANALYZE YOUR RETURN BEFORE THE I.R.S. DOES IT FOR YOU...

introducing . . .

THE TAX CONSULTANT™

(FOR ATARI 400/800 AND THE COMMODORE 64 COMPUTERS)

NOW YOU CAN HAVE COMPLETE TAX PREPARATION AT HOME!

only **$95.00** tax deductible!

and your return will be closely checked and analyzed before the I.R.S. ever sees it! THE TAX CONSULTANT™ is a sophisticated yet very friendly program for the novice user. THE TAX CONSULTANT™ will help you to calculate, prepare and print your return. But then, at your request, THE TAX CONSULTANT™ will check your return very closely and issue an itemized print-out indicating those items that in its judgment may be questioned. *No other tax preparation package offers this feature!*

- **Features** –
- On-screen Prompts
- Continuous tax/refund readout
- Fast, 100% machine language
- All required I.R.S. forms included
- Easy to follow tutorial in manual
- Prints all necessary forms
- Multi-color screens
- Inexpensive updates

Plus!

A TAX PLANNING MODULE THAT, WHEN USED WITH THE ANALYSIS MODULE, WILL PROVIDE YOU WITH A YEAR-ROUND TOOL TO PLOT YOUR TAX STRATEGIES

Dealer inquiries welcome!

Information

(707) 422-9591

Mail Orders to: **Morbius Software Co., Inc.
Dept. CP, P.O. Box 1702
Vacaville, CA 95696**

Name ______________________

Address ______________________

City, State ______________________

☐ CHECK/MONEY ORDER ZIP ______________

☐ VISA Interbank #(MC) ______________

☐ MASTERCARD Expires ______________

Card No. ______________________

Signature ______________________

Price $95.00 plus $4.00 shipping and handling. Calif. residents add 6% sales tax. Please allow two weeks for checks to clear.
System Requirements: Atari 400/800 with 48K & one disk drive; Commodore 64 with one disk drive. Printer recommended.
ATARI 400/800 is a trademark of Atari, Inc.
Commodore 64 is a trademark of Commodore Electronics, Ltd.
THE TAX CONSULTANT is a trademark of GMG Associates

VISA master charge

MASTERCARD / VISA CALL TOLL-FREE!

CALIF **1-800-821-6771** NATION **1-800-821-6770**

Telephone operators standing by Mon.-Fri. 8 am to 5 pm, Saturdays 9 am to 1 pm, PST

CHARACTER GRAPHICS continued from page 60
which looks somewhat like an upside-down U with upturns at the sides (See Figure 1, noting that each pixel to be displayed has its bit "on" and each pixel not to be displayed has its bit turned "off").

Step 3 — Calculate the byte value for each of the eight rows of your character as described in the definition of "bit" above, and make a list of the values next to each row on your grid. For the omega character, the decimal byte values are (from top to bottom): 0, 24, 60, 36, 36, 165, 231, and 0.

Step 4 — Decide which existing character you wish to replace with the new character. In this example we will replace the ampersand ("&") character, one of the least used punctuation marks.

Step 5 — Find the location in the OS ROM character set of the character to be replaced. Use Table 9.6 on page 55 of the *Atari BASIC Reference Manual* to find the position of the desired character. Write this number down at the beginning of the list of byte values that you recorded in Step 3. For the ampersand, this number is six.

Step 6 — Place these nine numbers in a DATA statement. In our case, the BASIC line looks like this:

```
DATA 6,0,24,60,36,36,165,231,0
```

Now go back and repeat Steps 1 through 6 for each character that you wish to redefine. Play around a little with some shapes you would like to create. When you are ready to see them displayed on the screen, go on to Step 7 below.

Step 7 — "Download" the standard character set into RAM memory. We want to replace some of the characters (or all of them, if you've been really creative) with our own designs, but we can't change bytes of ROM memory. The solution? Move the entire standard character set into RAM where it is easily accessed and modified. Since there are 128 characters in a set and each character consists of eight bytes, we must move the contents of 1024 bytes (one kilobyte). To do this, PEEK at the value in ROM and POKE it into RAM. First, we use memory location 106 to find out how many pages of RAM are in your computer:

```
CH=PEEK(106)
```

Next, subtract four (the number of pages in a one-kilobyte-long character set) and multiply the result by 256 to find the exact address of the beginning of our memory area:

```
CH=(CH-4)*256
```

Now use a FOR/NEXT loop to transfer the standard character set to this new area. To do this, we need to know that the OS permanent character set begins at memory location 57344.

```
FOR LOOP=0 TO 1023
BYTE=PEEK(57344+LOOP)
POKE CH+LOOP,BYTE
NEXT LOOP
```

The standard set now resides in two places in memory (one in the OS ROM, another now in RAM). You can modify the set located in RAM.

Step 8 — Modify the relocated RAM character set to include your new characters. Using the DATA statement written previously, we modify the ampersand character by multiplying the position of this character by eight and adding the result to the variable CH, which is the starting address of the new character set. Another FOR/NEXT loop POKEs these eight bytes of character data into RAM at the proper location:

```
READ NUM
NUM=NUM*8
FOR LOOP=0 TO 7
READ BYTE
POKE CH+NUM+LOOP,BYTE
NEXT LOOP
```

If you modify more than one character you can, of course, use more FOR/NEXT loops to POKE the character(s) into memory.

Step 9 — Tell the OS to use the new character set instead of the standard set. This is accomplished with a single POKE command to location 756, which is known as the Character Base Address because the value found here points to the character set currently being used by the computer. All character sets must start on a new page of memory, or page boundary, which is evenly divisible by four. So divide the variable CH by 256 to find the page number of your new set and POKE this number into location 756:

```
CH=CH/256
POKE 756,CH
```

That's it! We are now ready to PRINT our new characters on the screen.

PROGRAM EXPLANATION

Listing 1 is an example of the use of redefined character sets in educational and scientific programming. It displays a schematic diagram of an electric circuit in Graphics Mode 2 and asks first-year physics questions in the text window.

Here is a line-by-line explanation of the sample program; many of the techniques used are adaptable to other programs and applications.

1000–1040: Dimension variables, set up title screen.
1050–1060: Jump to the subroutine that redefines the characters and POKE in the new location as in Step 9.
1070–1120: Draw the circuit on the screen.
1130–1170: Initial message; wait for [START] button to be pressed.
1180–1280: Display a question and input an answer.
1290: Is answer correct?
1300–1390: Wrong answer — display correct answer and go on.
1400–1490: Correct answer — increment number right and go on.
1500–1510: Go on to the next question.
1520–1560: Quiz over — display score and start endless loop.
1570–1710: Subroutine to redefine the character set.
1720–1900: DATA statements for new character shapes.
1910–2010: DATA statements that use the new characters

128 64 32 16 8 4 2 1

Figure 1

0	0	0	0	0	0	0	0
0	0	0	1	1	0	0	0
0	0	1	1	1	1	0	0
0	0	1	0	0	1	0	0
0	0	1	0	0	1	0	0
1	0	1	0	0	1	0	1
1	1	1	0	0	1	1	1
0	0	0	0	0	0	0	0
7	6	5	4	3	2	1	0

Bit #

Capital omega (Ω).

Figure 2

0	0	0	1	1	0	1	1
1	1	0	0	0	1	1	0
1	0	1	1	0	0	0	1
0	1	1	0	1	1	0	0
0	0	0	1	1	0	1	1
1	1	0	0	0	1	1	0
1	0	1	1	0	0	0	1
0	1	1	0	1	1	0	0
7	6	5	4	3	2	1	0

Bit #

Checkerboard pattern produces characters at left and right in ANTIC Mode 4.

to draw the circuit diagram.

2020–2220: DATA statements that contain the text of the questions, choices, correct answers, and explanations.

Table 1 shows which characters were redefined in this program and how they were changed.

There are several points worth remembering when you use redefined character sets in your programs. Notice that in line 1590 the step-back from the top of RAM was eight pages. Since the screen memory and display list reside at the top of RAM, and they occupy more than a page of memory, we stepped back two pages. Now, since we want to relocate the entire four-page (1K) character set, we step back another four pages. These additional two pages were required because a 1K character set must start on a page that is evenly divisible by four!

You may remember that in Graphics Modes 1 and 2, only 64 of the 128 characters in a set may be displayed at one time (see the *Atari BASIC Reference Manual*, pp. 46–47, for an explanation of this). If you are using one of these modes, the character set only needs to be 64 characters long, or two pages instead of four, and must begin on an even-numbered page. But since our sample program prints text in the Graphics Mode 0 text window, which uses all 128 characters, we moved the entire set. If you are confused, see Table 2, which shows that the amount of memory you must reserve depends on the graphics mode you are using. When in doubt, use a larger step-back to avoid conflicting with the display list or screen.

You may notice a long pause after the title is displayed. This results from the FOR/NEXT loop used to download the character set from ROM into RAM. Fortunately, you can use a ready-made, machine-language routine to accomplish the same thing in a fraction of the time. Replace lines 1610–1640 in Listing 1 with these:

```
1610 DIM MOVE$(32):RESTORE 1630
1620 FOR LOOP=1 TO 32:READ BYTE:MOVE$
     (LOOP)=CHR$(BYTE):NEXT LOOP
1625 JUNK=USR(ADR(MOVE$),CH,57344)
1630 DATA 104,104,133,215,104,133
1632 DATA 214,104,133,213,104,133
1634 DATA 212,162,4,160,0,177
1636 DATA 212,145,214,200,208,249
1638 DATA 230,213,230,215,202,208
1640 DATA 240,96
```

For those of you who are familiar with 6502 assembly language, Listing 2 provides the source code for this 32-byte subroutine. For those of you who know nothing about assembly language, the routine is fully usable as is, as long as the variable "CH" contains the memory address of the new character set area.

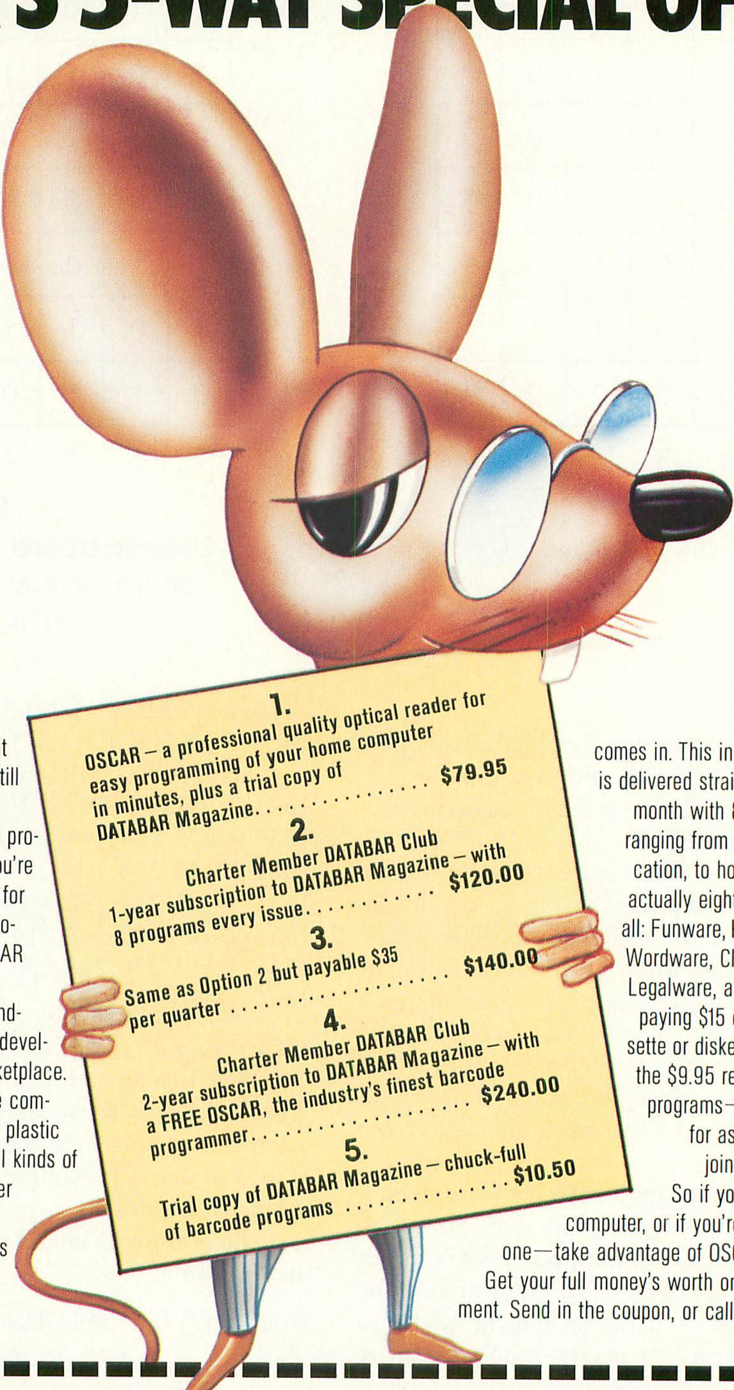
If you're using BASIC A+, replace lines 1610–1640 with the following:

```
1610 MOVE 57344,CH,1024
```

Since BASIC A+ has a reserved memory area for character sets and the like, you can set the variable "CH" to this address (see pg. L-1 of the manual). Whatever you do, do not use the PEEK(106)–4 method from Step 7, because BASIC A+ resides at the top of RAM. If you plan to redefine virtually every character in the set (for the particular graphics mode you're using), consider reserving an area of memory, placing the new character definitions there, and using the POKE to location 756 without ever downloading the standard set. That part of Step 7 becomes useless if none of the standard characters are to be used.

continued on page 65

OSCAR'S 5-WAY SPECIAL OFFER!



If you want to get more benefit out of your home computer and still avoid the purchase of expensive peripherals…if complex keyboard programming leaves you cold and you're tired of paying exhorbitant prices for every new cassette or diskette program—then you're ready for OSCAR and the DATABAR Club!

OSCAR is simply the finest hand-held optical barcode reader ever developed for the home computer marketplace. It works with virtually every home computer. And together with a simple plastic template, it allows you to enter all kinds of programs into your home computer in just a few minutes.

But hardware without programs is useless—and that's where the DATABAR Club comes in. This includes a magazine that is delivered straight to your home every month with 8 programs on subjects ranging from fun and games, to education, to home finances. There are actually eight different categories in all: Funware, Homeware, Healthware, Wordware, Classware, Scienceware, Legalware, and Genware. Instead of paying $15 or $30 or more for cassette or diskette programs, or paying the $9.95 retail price for DATABAR programs—you can receive them for as little as $1.25 each by joining the DATABAR Club.

So if you already have a home computer, or if you're thinking about buying one—take advantage of OSCAR's 5-way offer now. Get your full money's worth on your computer investment. Send in the coupon, or call for further information.

Please send me the Databar option(s) indicated.
☐ Option 1 ($79.95)†*†† ☐ Option 3 ($35.00 Q1)
☐ Option 2 ($120.00) ☐ Option 4 ($240.00) †
☐ Option 5 ($10.50)

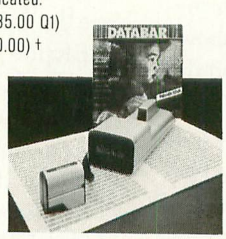
*30-day money-back guarantee. †Add $1.50 for handling. ††MN residents add 6% sales tax.

Enclosed Check # ____________

Money Order # ____________

Payable to Databar.

Or charge my Am Ex ☐ Visa ☐
MasterCard ☐
Carte Blanche ☐
Diners Club ☐



Card # ____________

Exp. Date ____________

Name ____________

Address ____________

City ________ State ____ Zip ____

Phone ____________

My Computer Model is ____________

DATABAR

Databar Corporation, 10202 Crosstown Circle • Eden Prairie, MN 55344 • Phone Orders: MN (612) 944-5700
Out-of-State: 1 (800) 672-2776

CHARACTER GRAPHICS continued from page 63

A problem may arise if you wish to redefine letters and numbers for a Graphics Mode 1 or 2 display and still wish to use the standard characters in the text window. To re-instruct the ANTIC processor to display the standard set, it is necessary to insert a display list interrupt (DLI) before the text window (if you're not familiar with DLI's, see ANTIC, June 1983, for an excellent tutorial). Once you have set up your graphics mode, use the following code:

```
DL = PEEK(560) + 256*PEEK(561)
FOR LOOP = 0 TO 10:READ BYTE: POKE 1536 + LOOP,
  BYTE:NEXT LOOP
DATA 72,169,224,141,10,212,
DATA 141,9,212,104,64
POKE 512,0:POKE 513,6
POKE DL + 14,135:REM — USE POKE DL + 24,134 IN
  GRAPHIC MODE ONE
POKE 54286,192
```

Listing 3 is the source code for this DLI service routine. It sets the character base address to point to the standard set on the line before the text window is displayed.

MULTICOLOR CHARACTERS

Now, on to multicolor characters, a little-known feature of your Atari. If you looked at Table 2, you noticed "ANTIC Modes 4 and 5." They don't have a BASIC mode number because they can't be called directly from BASIC like Graphics Modes 0, 1, and 2. To access them, you must modify the display list, a machine-language program that tells the ANTIC chip how to display data on the screen.

If you are unfamiliar with custom display lists, read "Display Lists Simplified" (ANTIC, February–March 1983). You do not need to completely understand display lists to implement multicolor characters, however, as you can use the ready-made routines provided below.

ANTIC mode 4 creates four-color characters that are the size of BASIC Mode 0 characters — 24 rows of 40 will fit on a screen. ANTIC Mode 5 doubles the height of these characters. Therefore, only 12 rows of 40 will fit on the screen.

The following code will set up a full ANTIC Mode 4 screen:

```
GRAPHICS 0
DL = PEEK(560) + 256*PEEK(561)
POKE DL + 3,68
FOR LOOP = 6 TO 28
POKE DL + LOOP,4
NEXT LOOP
```

And this routine will set up an ANTIC Mode 5 screen:

```
GRAPHICS 0
DL = PEEK(560) + 256*PEEK(561)
A = PEEK(559):POKE 559,0
POKE DL + 3,69
FOR LOOP = 6 TO 16
POKE DL + LOOP,5
NEXT LOOP
POKE DL + 17,65
POKE DL + 18,PEEK(560):POKE DL + 19,PEEK(561)
POKE 559,A
```

The POKE to location 559 completely shuts down the screen display while the display list is being modified. If this were not included, ANTIC might get confused by our changes to its program and display some garbage on the screen. This measure is not needed when setting up ANTIC Mode 4 because its display list is just as long as BASIC Graphics Mode 0's. If you know something about display lists, you will notice that the display list for ANTIC Mode 5 is shorter than most. It only requires 12 lines. Consequently, the original display list must be shortened while ANTIC is following it.

Character redefinition is basically the same for multicolor modes as for "normal" modes. Steps 1 through 9 remain the same, but the process of designing new shapes is different.

Each character is only four pixels or dots wide, and each

continued on next page

Figure 2a

NORMAL VIDEO

0	1	2	3
3	0	1	2
2	3	0	1
1	2	3	0
0	1	2	3
3	0	1	2
2	3	0	1
1	2	3	0

Numbers are color registers;
0 = background color.

Figure 2b

INVERSE VIDEO

0	1	2	4
4	0	1	2
2	4	0	1
1	2	4	0
0	1	2	4
4	0	1	2
2	4	0	1
1	2	4	0

Numbers are color registers;
0 = background color.

pixel can be any one of four colors. In each of the eight bytes that make up a character, two bits determine the color of a pixel:

00 = Background color
01 = Color register zero
10 = Color register one
11 = Color register two (or color register three, if the character is displayed in inverse video)

Figure 2 is an example of a four-color checkerboard character. Listing 4 is an example of animation with a multicolor character mode. In this program, I've kept the last line of the screen in BASIC Graphics Mode 0 for the display of various messages. The animation is not flicker-free, of course, but it is easy to program and uses several characters. Note that this program incorporates the machine-language routine for moving the character set.

Table 1

POSITION IN CHARACTER SET	ASCII VALUE	ORIGINALLY	REDEFINED TO
1	33	!	⊥
2	34	"	L
3	35	#	⊣
4	36	$	>
5	37	%	¹
6	38	&	Ω
7	39	'	⅃
10	42	*	≡
26	58	:	T
27	59	;	⊢
28	60	<	³
32	64	@	²
59	91	[	A (boxed)
60	92	\	—
61	93	]	|
62	94	^	┐
63	95	_	┌

Redefined characters used in Listing 1.

Table 2

GRAPHICS MODE	MEMORY RESERVATION*
0	4 Pages
1	2 Pages (4 with text window)
2	2 Pages (4 with text window)
ANTIC Mode 4	4 Pages
ANTIC Mode 5	4 Pages

*Not including screen, display list memory.

Memory reservation required for character sets.

Requires 16K RAM

Listing 1

```
999 REM ANTIC MAGAZINE
1ØØØ REM *** CHARACTER SET EXAMPLE #1
1Ø1Ø GRAPHICS 2:POKE 752,1:POKE 82,1:P
OKE 83,38:PRINT CHR$(125):REM - SET UP
 GRAPHICS MODE
1Ø2Ø DIM LINE$(2Ø),QU$(38),ANS1$(38),A
NS2$(38),AN$(1),EX$(38),TAN$(1Ø)
1Ø3Ø POSITION 6,2:PRINT #6;"ELECTRIC":
POSITION 6,4:? #6;"circuits"
1Ø4Ø PRINT "     Character Set Sample
Program":PRINT "            by Chris Ch
abris"
1Ø5Ø GOSUB 157Ø:REM - REDEFINE CHARACT
ERS SUBROUTINE
1Ø6Ø POKE 756,CH/256:REM - ACTIVATE NE
W CHARACTER SET
1Ø7Ø REM - NOW PRINT OUT THE CIRCUIT O
N THE SCREEN FROM DATA STATEMENTS
1Ø8Ø RESTORE 192Ø
1Ø9Ø FOR LOOP=Ø TO 9
11ØØ READ P,LINE$
111Ø POSITION P,LOOP:PRINT #6;LINE$;
112Ø NEXT LOOP
113Ø REM - PRINT INITIAL MESSAGE
114Ø PRINT CHR$(125)
115Ø PRINT "                 Press START"
:PRINT "     when you are ready to begi
n."
116Ø POKE 53279,Ø
117Ø IF PEEK(53279)<>6 THEN 117Ø
118Ø REM - NOW WE'RE READY TO BEGIN T
HE QUESTIONS
119Ø RESTORE 2Ø2Ø
12ØØ FOR LOOP=1 TO 5
121Ø PRINT CHR$(125);
122Ø READ QU$,ANS1$,ANS2$,AN$,EX$,TAN$
123Ø PRINT QU$:PRINT ANS1$:PRINT ANS2$
124Ø FOR LOOP2=1 TO 5Ø:SOUND Ø,1ØØ,1Ø,
8:NEXT LOOP2:SOUND Ø,Ø,Ø,Ø
125Ø OPEN #1,4,Ø,"K:"
126Ø POKE 7Ø2,64:REM - KEYBOARD TO ALL
 CAPITALS
127Ø GET #1,AN
128Ø CLOSE #1
129Ø IF AN=ASC(AN$) THEN 14ØØ:REM - CO
RRECT ANSWER
13ØØ FOR LOOP2=1 TO 1ØØ:SOUND Ø,25Ø,1Ø
,8:NEXT LOOP2:SOUND Ø,Ø,Ø,Ø
131Ø PRINT CHR$(125);
132Ø PRINT QU$
```

```
133Ø PRINT "The correct answer is ";TA
N$;".":PRINT EX$
134Ø PRINT "Press RETURN to go on.";
135Ø OPEN #1,4,Ø,"K:"
136Ø GET #1,AN
137Ø IF AN<>155 THEN 136Ø
138Ø CLOSE #1
139Ø GOTO 15ØØ:REM - NEXT QUESTION
14ØØ REM - CORRECT ANSWER
141Ø FOR LOOP2=1 TO 75:SOUND Ø,15Ø,1Ø,
8:NEXT LOOP2:SOUND Ø,Ø,Ø,Ø
142Ø PRINT CHR$(125)
143Ø PRINT "                You are correct
."
144Ø PRINT "            Press RETURN to go
 on."
145Ø OPEN #1,4,Ø,"K:"
146Ø GET #1,AN
147Ø IF AN<>155 THEN 146Ø
148Ø CLOSE #1
149Ø RIGHT=RIGHT+1
15ØØ REM - NEXT QUESTION
151Ø NEXT LOOP
152Ø PRINT CHR$(125)
153Ø PRINT "  You have completed this
short quiz"
154Ø PRINT "      with a score of ";(RI
GHT/5)*1ØØ;" per cent."
155Ø GOTO 155Ø:REM - USER MUST PRESS B
REAK OR SYSTEM RESET TO EXIT
156Ø END
157Ø REM ** REDEFINE CHARACTER SET
158Ø REM - FIRST, RESERVE MEMORY
159Ø CH=PEEK(1Ø6)-8:CH=CH*256:REM - S
TARTING ADDRESS OF NEW CHARACTER SET A
REA
16ØØ REM - NOW RELOCATE THE OLD SET
161Ø FOR LOOP=Ø TO 1Ø23
162Ø BYTE=PEEK(57344+LOOP)
163Ø POKE CH+LOOP,BYTE
164Ø NEXT LOOP
165Ø REM - NOW WE'RE READY TO POKE IN
 THE NEW CHARACTER DEFINITIONS
166Ø RESTORE 173Ø
167Ø READ NUM:IF NUM=-1 THEN RETURN :R
EM - IF DONE, GO BACK
168Ø NUM=NUM*8:REM - OFFSET INTO NEW
CHARACTER SET MEMORY
169Ø FOR LOOP=Ø TO 7:READ BYTE:POKE CH
+NUM+LOOP,BYTE
17ØØ NEXT LOOP
171Ø GOTO 167Ø:REM - READY FOR ANOTHER
 CHARACTER
172Ø REM - CHARACTER REDEFINITIONS
173Ø DATA 1,24,24,24,255,255,Ø,Ø,Ø
174Ø DATA 2,24,24,24,31,31,Ø,Ø,Ø
175Ø DATA 3,24,24,24,248,248,24,24,24
176Ø DATA 4,24,12,6,3,3,6,12,24
177Ø DATA 5,64,192,64,64,224,Ø,Ø,Ø
178Ø DATA 6,Ø,24,6Ø,36,36,165,231,Ø
179Ø DATA 7,24,24,24,248,248,Ø,Ø,Ø
18ØØ DATA 1Ø,255,Ø,6Ø,Ø,255,Ø,6Ø,Ø
181Ø DATA 26,Ø,Ø,Ø,255,255,24,24,24
182Ø DATA 27,24,24,24,31,31,24,24,24
183Ø DATA 28,224,32,96,32,224,Ø,Ø,Ø
184Ø DATA 32,224,32,224,128,224,Ø,Ø,Ø
185Ø DATA 59,126,129,189,165,189,165,1
29,126
186Ø DATA 6Ø,Ø,Ø,Ø,255,255,Ø,Ø,Ø
187Ø DATA 61,24,24,24,24,24,24,24,24
188Ø DATA 62,Ø,Ø,Ø,248,248,24,24,24
189Ø DATA 63,Ø,Ø,Ø,31,31,24,24,24
19ØØ DATA -1
191Ø REM - DATA FOR SCREEN
192Ø DATA 2,3.0A
193Ø DATA 1,_\\[\\:\\\:\\\^
194Ø DATA 1,]        ]    ]    ]
195Ø DATA 1,*        $    $    $
196Ø DATA 1,*24V     $R   $R   $R
197Ø DATA 1,*        $ %  $ @  $
198Ø DATA 1,]        ]    ]    ]
199Ø DATA 1,"\\\\\\!\\\\!\\\'
2ØØØ DATA 1,R =20& AND R =30&
2Ø1Ø DATA 2,%
2Ø2Ø REM - QUESTIONS FOLLOW
2Ø3Ø DATA 1. What does the symbol & r
epresent?
2Ø4Ø DATA (a) coulombs          (b) weber
s
2Ø5Ø DATA (c) ohms              (d) newto
ns
2Ø6Ø DATA C,Ohms measure resistance.,
ohms
2Ø7Ø DATA 2. The resistors are connect
ed in ...
2Ø8Ø DATA (a) series            (b) paral
lel
2Ø9Ø DATA (c) sequence          (d) inver
sion
21ØØ DATA B,They are each in separate
branches.,parallel
211Ø DATA 3. What is the equivalent re
sistance?
212Ø DATA (a) Ø.13 &            (b) 8.Ø
&
213Ø DATA (c) 58 &              (d) 72 &
214Ø DATA B,Eq. Resistance=Voltage/Cur
rent (24/3),8.0 &
215Ø DATA 4. The power supplied is ...
216Ø DATA (a) 22Ø w             (b) 19Ø w
atts
217Ø DATA (c) 24 w              (d) 72 w
218Ø DATA D,Power in a ciruit=Voltage
X Current.,72 w
219Ø DATA 5. The current in resistor 1
 is ...
22ØØ DATA (a) Ø.83 a            (b) 1.5 a
221Ø DATA (c) 3.Ø a             (d) 1.2 a
222Ø DATA D,Current=Voltage/Resistance
 (24/2Ø),1.2 a
```

TYPO TABLE

Variable checksum = 393ØØ2

Line num range	Code	Length
999 - 1Ø7Ø	AI	567
1Ø8Ø - 119Ø	BX	3Ø1

continued on next page

```
12ØØ    - 131Ø     JB    433
132Ø    - 143Ø     QW    365
144Ø    - 155Ø     EM    356
156Ø    - 167Ø     DQ    367
168Ø    - 179Ø     TR    387
18ØØ    - 191Ø     JL    348
192Ø    - 2Ø3Ø     XZ    287
2Ø4Ø    - 215Ø     KI    473
216Ø    - 222Ø     XX    277
```

Listing 2

```
LISTING TWO
-----------
; Character set movement subroutine
; Stored in MOVE$
; Call from Atari BASIC with:
;   JUNK=USR(ADR(MOVE$),CH,57344)
ROM     EQU    $ØØD4    ; Address of
ROM character set - should be $EØØØ
RAM     EQU    $ØØD6    ; New location
 of character set from variable "CH"
INIT    PLA             ; Discard numb
er of parameters from BASIC
        PLA
        STA    RAM+1
        PLA
        STA    RAM      ; Store the ad
dress of the RAM set from the stack
        PLA
        STA    ROM+1
        PLA
        STA    ROM      ; Do the same
for the ROM set
        LDX    #Ø4      ; 4 pages of m
emory to do
LOOP1   LDY    #ØØ      ; Zero the cou
nter of bytes done
LOOP2   LDA    (ROM),Y  ; Get byte fro
m ROM set
        STA    (RAM),Y  ; And store it
 in the RAM set area
        INY
        BNE    LOOP2    ; If not done
with one page, do the next byte
        INC    RAM+1
        INC    ROM+1    ; Add one page
        DEX             ; Reduce count
er of pages left to do
        BNE    LOOP1    ; If not zero,
 go back and do the next page
        RTS             ; All done, so
 go back to BASIC
```

Listing 3

```
LISTING THREE
-------------
; Display List Interrupt service routi
ne
; Changes Character Base Pointer befor
e text window is displayed
WSYNC   EQU    $D4ØA    ; Wait for hor
izontal synchronization register
CHBASE  EQU    $D4Ø9    ; Character Ba
se Pointer register
        PHA             ; Save accumul
ator for return to normal processing
        LDA    #$EØ     ; Hibyte of RO
M character set starting address
        STA    WSYNC    ; Wait for pro
per synchroniztion
        STA    CHBASE   ; Now put the
new character base into ANTIC
        PLA             ; Restore accu
mulator
        RTI             ; All done, so
 go back to noraml execution
```

Listing 4

```
999 REM ANTIC MAGAZINE
1ØØØ REM *** CHARACTER SET EXAMPLE #2
1Ø1Ø GRAPHICS Ø:POKE 752,1:PRINT CHR$(
125):SETCOLOR 2,Ø,Ø
1Ø2Ø DIM DRAW$(13),UNDRAW$(13)
1Ø3Ø GOSUB 126Ø:REM - MODIFY DISPLAY L
IST SUBROUTINE
1Ø4Ø GOSUB 131Ø:REM - MODIFY CHARACTER
 SET SUBROUTINE
1Ø5Ø REM - DRAW THE SCREEN WITH OUR SH
APES
1Ø6Ø LET 1Ø6Ø LET DRAW$="┘↓←┘↓←┘↓←←♥▌┤
":LET UNDRAW$=" ↓← ↓← ↓←←    "
1Ø7Ø POSITION 2,22:PRINT "------------
--------------------------";:REM - LAUNC
H PAD
1Ø8Ø POKE 756,CH/256
1Ø9Ø POSITION 2Ø,18:? DRAW$
11ØØ FOR LOOP=9 TO Ø STEP -1
111Ø POSITION 2Ø,23:? LOOP;
112Ø FOR LOOP2=1 TO 2ØØ:NEXT LOOP2
113Ø NEXT LOOP
114Ø POSITION 15,23:PRINT "BLAST  OFF!
";
115Ø FOR LOOP=1 TO 3ØØ:NEXT LOOP
116Ø POSITION 12,23:PRINT "SHE'S LIFTI
NG OFF";
117Ø FOR LOOP=17 TO Ø STEP -1
118Ø POSITION 2Ø,LOOP+1:? UNDRAW$;
119Ø POSITION 2Ø,LOOP:? DRAW$
12ØØ FOR LOOP2=1 TO 2*LOOP:NEXT LOOP2
121Ø NEXT LOOP
122Ø POSITION 1Ø,23:PRINT "MISSION ACC
OMPLISHED!";
123Ø POSITION 2Ø,Ø:? UNDRAW$
124Ø GOTO 124Ø:REM - USER MUST PRESS B
REAK OR SYSTEM RESET TO EXIT
125Ø STOP
126Ø REM - MODIFY THE DISPLAY LIST
127Ø DL=PEEK(56Ø)+256*PEEK(561)
128Ø POKE DL+3,68:REM - ANTIC MODE 4
129Ø FOR LOOP=6 TO 27:POKE DL+LOOP,4:N
EXT LOOP:REM - CHANGE ALL SCREEN LINES
13ØØ RETURN
131Ø REM - DOWNLOAD AND MODIFY THE CHA
RACTER SET
132Ø CH=PEEK(1Ø6)-8:CH=CH*256
133Ø DIM MOVE$(32):RESTORE 135Ø:FOR LO
OP=1 TO 32:READ BYTE:MOVE$(LOOP)=CHR$(
BYTE):NEXT LOOP
134Ø JUNK=USR(ADR(MOVE$),CH,57344)
135Ø DATA 1Ø4,1Ø4,133,215,1Ø4,133
136Ø DATA 214,1Ø4,133,213,1Ø4,133
```

```
1370 DATA 212,162,4,160,0,177
1380 DATA 212,145,214,200,208,249
1390 DATA 230,213,230,215,202,208
1400 DATA 240,96
1410 RESTORE 1480
1420 READ NUM:IF NUM=-1 THEN RETURN
1430 NUM=NUM*8
1440 FOR LOOP=0 TO 7:READ BYTE:POKE CH
+NUM+LOOP,BYTE
1450 NEXT LOOP
1460 GOTO 1420
1470 REM - CHARACTER REDEFINITIONS
1480 DATA 64,2,10,42,42,170,170,0,0
1490 DATA 65,128,160,168,168,170,170,0
,0
1500 DATA 66,190,255,255,255,255,255,2
0,85
1510 DATA 67,190,190,190,190,190,190,1
90,190
1520 DATA 68,0,0,0,20,20,40,170,190
1530 DATA -1
```

TYPO TABLE

Variable checksum = 280222

Line num range	Code	Length
999 - 1100	UA	488
1110 - 1220	YX	332
1230 - 1340	AD	453
1350 - 1460	JP	263
1470 - 1530	YJ	215

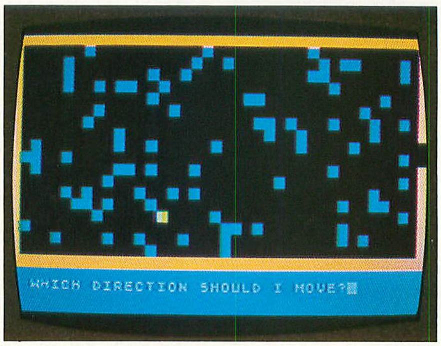
Gantlet

by STEPHEN GROLL

Requires 16K cassette, 24K disk

To run **The Gantlet**, you must pass through twelve rooms in sequence. There is only one exit from each room. Your challenge is to find each of them.

Maneuver your player (a colored square) through the gantlet by typing U for up, D for down, L for left and R for right. Then respond to the prompt "How Many Steps" with the number of steps you want to take. Watch your player carefully as it responds to contact with the walls and other objects. You can catch subtle clues by noticing the way it moves or reacts.



After you've figured out the secret of each room, see how quickly you can run the gantlet. As you move through the rooms, try to discover the step-saving and time-saving moves that will enable you to move through it more quickly. The computer displays your time after you exit the last room. Good luck!

```
1 REM ** THE GANTLET ***
2 REM * BY STEVE GROLL *
3 REM * ANTIC MAGAZINE *
4 REM * FEBUARY, 1984  *
5 REM ****************
10 GRAPHICS 18:POSITION 5,2:? #6;"THE GANTLET":POSITION 10,5:? #6;"BY":POSITION 5,8:? #6;"steve groll"
20 FOR T=1 TO 1000:NEXT T:DIM D$(1):TRAP 20000
50 POKE 20,0:POKE 19,0:POKE 18,0:LC=0:RR=0:UR=0
100 GRAPHICS 3:COLOR 1:GOSUB 19000:COLOR 4:PLOT 39,9:PLOT 39,10
140 FOR L=1 TO 100:XX=(37*RND(1))+1:YY=(17*RND(1))+1:COLOR 3:PLOT XX,YY:NEXT L
160 X=INT(20*RND(1))+1:Y=INT(18*RND(1))+1:COLOR 2:PLOT X,Y:C=2
200 GOSUB 20000:IF W<>0 THEN GOSUB 19200:GOTO 100
```

Stephen Groll is pastor of the First Baptist Church of Sonoma, CA. He purchased an Atari 400 about a year ago for use as an educational tool and for his own amusement. He is a self-taught programmer.

```
300 GRAPHICS 3:COLOR 1:SETCOLOR 0,5,2:GOSUB 19000:X=1:Y=9:G=0
310 COLOR 1:PLOT 1,6:DRAWTO 38,6:PLOT 1,13:DRAWTO 38,13:SETCOLOR 1,2,8
320 PLOT 12,1:DRAWTO 12,18:PLOT 27,1:DRAWTO 27,18:OPN=0
330 COLOR 3:PLOT X,Y:GOSUB 20000:IF W=2 THEN X=33:Y=9:GOTO 397
335 IF Y=13 AND OPN=0 THEN Y=14:GOTO 397
340 IF Y=19 AND W=1 THEN Y=9:OPN=1:GOTO 397
360 IF Y=6 AND OPN=1 THEN X=X+2:Y=4:GOSUB 19100:W=0:IF X=12 OR X=27 THEN X=X+1:GOTO 330
365 IF (X=40 OR X=39) AND Y=4 AND W=0 THEN X=38:Y=7:COLOR 2:PLOT 17,19:DRAWTO 12,19:DRAWTO 12,6:DRAWTO 39,6:G=1
367 IF G=1 THEN DRAWTO 39,19:DRAWTO 21,19:COLOR 0:DRAWTO 20,19:DRAWTO 18,19:G=0:COLOR 2:PLOT X,Y:RET=1
380 IF X=28 AND Y=7 THEN X=19:Y=9:GOTO 397
383 IF X=28 AND Y=12 THEN X=19:Y=17:GOTO 397
```

continued on page 72



Put yourself in the pilot's seat of a Piper 181 Cherokee Archer for an awe-inspiring flight over realistic scenery from New York to Los Angeles. High speed color-filled 3D graphics will give you a beautiful panoramic view as you practice takeoffs, landings, and aerobatics. Complete documentation will get you airborne quickly even if you've never flown before. When you think you're ready, you can play the World War I Ace aerial battle game. Flight Simulator II features include ■ animated color 3D graphics ■ day, dusk, and night flying modes ■ over 80 airports in four scenery areas: New York, Chicago, Los Angeles, Seattle, with additional scenery areas available ■ user-variable weather, from clear blue skies to grey cloudy conditions ■ complete flight instrumentation ■ VOR, ILS, ADF, and DME radio equipped ■ navigation facilities and course plotting ■ World War I Ace aerial battle game ■ complete information manual and flight handbook.

See your dealer . . .

or write or call for more information. For direct orders please add $1.50 for shipping and specify UPS or first class mail delivery. American Express, Diner's Club, MasterCard, and Visa accepted.

Order Line: 800/637-4983

subLOGIC Corporation
713 Edgebrook Drive
Champaign IL 61820
(217) 359-8482 Telex: 206995

THE GANTLET continued from page 70

```
385 IF X=38 AND Y=12 THEN X=34:Y=17:GOTO 397
386 IF X=38 AND Y=7 THEN X=33:Y=9:GOTO 397
388 IF W=0 AND Y=19 THEN 500
390 IF W=1 THEN 398
396 GOTO 330
397 GOSUB 19100:GOTO 330
398 GOSUB 19200:GOTO 300
500 GRAPHICS 3:SETCOLOR 1,9,6:SETCOLOR 2,0,0:C=1:COLOR 3:PLOT 38,14:PLOT 38,17:PLOT 36,16:PLOT 36,18:RET=0
510 COLOR 2:GOSUB 19000:Y=1:PLOT 19,9
515 COLOR 1:PLOT X,Y:GOSUB 20000
520 IF X=19 AND Y=9 THEN PLOT 39,16:DRAWTO 39,18:GOSUB 19100:GOTO 515
530 IF W<>0 THEN X=19:GOSUB 19200:GOTO 510
900 GRAPHICS 3:COLOR 2:SETCOLOR 4,4,0:SETCOLOR 1,0,0:GOSUB 19000:X=1
920 COLOR 1:PLOT X,Y:GOSUB 20000:IF Y>0 AND Y<9 AND X=39 THEN 1000
930 GOSUB 19200:X=19:Y=9:GOTO 920
1000 GRAPHICS 3:COLOR 1:Y=2:A=0:B=0:C=39:D=0:RET=0
1010 FOR L=0 TO 19:PLOT A,B:DRAWTO C,D:B=B+1:D=D+1:NEXT L
1020 COLOR 4:PLOT 0,9:DRAWTO 39,9:PLOT 0,10:DRAWTO 39,10:PLOT 19,0:DRAWTO 19,18:PLOT 20,18:DRAWTO 20,0
1030 PLOT 17,7:DRAWTO 22,7:PLOT 17,8:DRAWTO 22,8:PLOT 17,11:DRAWTO 22,11:PLOT 17,12:DRAWTO 22,12
1040 COLOR 3:X=19:Y=9:PLOT X,Y
1045 IF LC=5 AND RR=4 AND UR=1 THEN COLOR 4:PLOT 19,19:PLOT 20,19
1046 IF LC=5 THEN COLOR 1:PLOT 0,9:PLOT 0,10
1047 IF RR=4 THEN COLOR 1:PLOT 39,9:PLOT 39,10
1048 IF UR=1 THEN COLOR 1:PLOT 19,0:PLOT 20,0
1050 GOSUB 20000:IF W=1 THEN COLOR 1:PLOT X,Y:GOSUB 19200:GOTO 1040
1060 IF X=0 THEN 2000
1070 IF X=39 THEN 3000
1080 IF Y=0 THEN 3100
1090 IF Y=19 THEN 3300
2000 GRAPHICS 3:COLOR 1:SETCOLOR 0,4,0:SETCOLOR 4,7,2:SETCOLOR 2,0,0:GOSUB 19000
2010 PLOT 0,13:DRAWTO 38,13:COLOR 0:PLOT 39,9:PLOT 39,10:COLOR 2:PLOT 19,16:X=38:COLOR 3:PLOT X,Y:LC=0
2020 COLOR 2:PLOT 9,7:PLOT 29,7
2030 GOSUB 20000:IF W=1 OR (W=3 AND LC<2) THEN GOSUB 19200:GOTO 1000
2035 IF X=39 THEN 1000
2040 IF (LC=0 OR LC=1) AND W=2 THEN LC=LC+1:COLOR 3:PLOT X,Y:Y=5:PLOT X,Y:GOTO 2090
2050 IF (LC=2 OR LC=3) AND W=3 THEN LC=LC+1
2055 IF LC=3 THEN 2090
2060 IF LC=4 AND W=3 THEN PLOT 19,13:PLOT 20,13:GOTO 2090
2070 COLOR 3:PLOT X,Y:LC=5
2090 GOSUB 19100:GOTO 2030
2586 IF RB=2 AND LB=3 THEN PLOT 18,19:DRAWTO 20,19
3000 GRAPHICS 3:COLOR 1:GOSUB 19000:COLOR 0:PLOT 0,9:PLOT 0,10
3010 COLOR 1:PLOT 0,7:DRAWTO 39,7:COLOR 2:SETCOLOR 1,4,2:PLOT 9,4:PLOT 29,4:PLOT 19,14
3020 X=1:DO=0:RR=0
3030 COLOR 3:PLOT X,Y:GOSUB 20000:IF DO=0 AND X=19 AND Y=14 THEN Y=16:COLOR 1:PLOT 0,9:PLOT 0,10:DO=1:GOTO 3095
3040 IF DO=1 AND X=19 AND Y=14 THEN Y=16:PLOT 0,9:PLOT 0,10:DO=0:GOTO 3095
3050 IF DO=1 AND Y=19 THEN Y=1:GOTO 3095
3060 IF Y=0 THEN Y=18:GOTO 3095
3065 IF X=0 AND W=0 THEN 1000
3070 IF W=2 THEN COLOR 3:PLOT X,Y:X=19:RR=RR+1:GOTO 3095
3080 IF W=3 THEN COLOR 1:PLOT X,Y:X=19:RR=RR+1:GOTO 3095
3085 IF W=1 AND Y=4 THEN RR=0
3090 X=19:Y=16:GOSUB 19200:GOTO 3030
3095 GOSUB 19100:GOTO 3030
3100 GRAPHICS 3:COLOR 2:SETCOLOR 1,0,0:SETCOLOR 4,12,2:GOSUB 19000:COLOR 1:PLOT 15,18:DRAWTO 15,11
3110 PLOT 16,13:DRAWTO 23,13:DRAWTO 23,8:PLOT 20,4:DRAWTO 33,4:PLOT 30,5
3120 DRAWTO 30,15:PLOT 31,7:DRAWTO 36,7:PLOT 36,3:DRAWTO 36,9:PLOT 34,18:DRAWTO 34,13:PLOT 26,11:DRAWTO 26,18
3130 PLOT 22,18:DRAWTO 22,16:DRAWTO 18,16:PLOT 25,1:DRAWTO 25,3
3140 PLOT 17,1:DRAWTO 17,8:PLOT 14,16:DRAWTO 9,16:PLOT 11,15:DRAWTO 11,9:PLOT 6,18:DRAWTO 6,14:DRAWTO 3,14
3150 PLOT 1,5:DRAWTO 12,5:DRAWTO 12,2:PLOT 5,1:DRAWTO 5,3:PLOT 16,7:DRAWTO 14,7
3160 PLOT 1,17:PLOT 2,17:PLOT 3,6:DRAWTO 3,10:DRAWTO 7,10:PLOT 8,9:DRAWTO 8,13:PLOT 6,2:DRAWTO 7,2
3170 COLOR 0:PLOT 19,19:PLOT 20,19:COLOR 2:PLOT 2,2:Y=18:COLOR 3:PLOT X,Y:RET=1:SEE=3:UR=0
3172 IF SEE=3 THEN SETCOLOR 0,2,8:SEE=0:FOR T=1 TO 1000:NEXT T:SETCOLOR 0,12,2
3175 GOSUB 20000:IF X=2 AND Y=2 THEN COLOR 0:PLOT X,Y:X=34:Y=2:COLOR 3:PLOT X,Y:GOSUB 19100:SEE=2:W=0:UR=1
```

SOLUTIONS (USE ONLY IF YOU GET STUCK)

Please note: The Gantlet is not easy. Therefore, we're providing solutions to all of the rooms but the first. Try to figure out the puzzles on your own first, but if all else fails read these clues one at a time.

Room One is so easy that even the ANTIC staff figured it out. We'll leave this one to you.

Room Two is a bit tricker. First, move down until you hit the bottom wall; continue to move down until you hit the next bottom wall. You'll find yourself back where you started. Next, move up until you hit the top wall, and you'll be in the top left box. Move to the right edge of this box, then move down and hit the bottom wall. Repeat this procedure twice, and you'll find yourself in the right-middle box; at this point, you'll see four boxes enclosed in a golden rectangle. Move to the lower left-hand corner of the box, and the way out will become clear.

To find the exit from **Room Three**, first move down 10, then down 8. Next, move right 18, up 1, and right 3. That's it!

Room Four has an invisible door in the upper half of the right vertical wall. Find it and you will be in the fifth room.

Room Five has open corridors at the top and on either side. Each one leads to a number of rooms. Solve all three and the bottom of Room Five will open.

Room Six is at the end of the left corridor. First hit the green square nearest the entrance, then hit the green square at the far left. It will turn black. Hit it again, then hit the first square once more. A door will open to give you access to the last square. Hit it and exit.

Room Seven is at the end of the right corridor. Hit the red square closest to the bottom of the screen. The entrance door will close, but an invisible door will open in the bottom wall. Hit the bottom wall. Then hit the two squares *twice each* in any combination. Hit any wall and your player will appear at the bottom of the screen. Hit the red square again to open the door and exit.

Room Eight, at the end of Room Five's top corridor, contains a maze that appears once every three moves and then becomes invisible. Hit the square at the upper-left corner without hitting the walls. Your player will appear in the upper-right corner. Move back out of the maze.

Room Nine at the end of Room Five's bottom corridor, contains two white squares. You have two moves to hit one of them. If you do not, another white square appears. When you've hit all of the white squares another square will appear, blink a few times, and then vanish. You must hit the invisible square within four moves, or it will move to another place on the screen. After you've found and hit the invisible square four times, the door to the next room will open.

Room Ten contains two yellow bars. Hit the right bar either three of four times, then hit the left bar five or six times, and a door will open.

In **Room Eleven**, there are two light blue squares and two dark brown squares. One of each is surrounded by a green wall. Hit one of the exposed squares, and your player will turn the color of that square. If you hit the blue square, hit the bottom of the wall that surrounds the brown square and you will be given access to the square. Hit it, then hit the other brown square. Next, hit the left side of the wall that surrounds the other blue square. Once you've hit all four squares in any order, a door will open to the last room.

Room Twelve contains two black bars and one yellow bar. Touch the yellow bar and your player will turn yellow. Next hit the left black bar. Notice that the yellow bar drops one space. You must hit the yellow bar *before* you hit the black bars (each time). Hit the left black bar two more times, for a total of three hits. Then hit the right black bar twice. Hit the left bar once, then hit the right bar two more times. Next, hit the left bar twice. By this time, the yellow bar should be in the bottom wall. Hit the left bar once, then hit the right bar once, and the yellow bar will rise, leaving a gap in the wall. Exit from here and your time will be displayed.

```
3200 SEE=SEE+1:IF W<>0 THEN UR=0:GOSUB
 19200:GOTO 1000
3205 IF Y=19 THEN 1000
3210 GOTO 3172
3300 GRAPHICS 3:COLOR 3:SETCOLOR 2,15,
4:SETCOLOR 0,0,14:SETCOLOR 1,4,2:GOSUB
 19000:PLS=12:SQ=0
3310 Y=1:MM=0:RET=1:C=2
3312 FOR L=1 TO PLS
3315 X1=INT(38*RND(1))+1:Y1=INT(18*RND
(1))+1
3320 LOCATE X1,Y1,WW
3330 IF WW<>0 THEN 3315
3340 COLOR 1:PLOT X1,Y1:GOSUB 19400:SQ
=SQ+1:NEXT L
3345 IF SQ=0 THEN SQ=4:GOTO 3415
3350 COLOR 2:PLOT X,Y:GOSUB 20000:MM=M
M+1
3360 IF W=0 AND SQ<>0 AND MM=2 THEN PL
S=1:MM=0:GOTO 3312
3370 IF W=1 THEN GOSUB 19300:SQ=SQ-1:M
M=0:GOTO 3345
3380 IF W=3 THEN COLOR 3:PLOT 0,5:PLOT
 0,6:X=19:Y=9:PLS=4:GOTO 3312
3390 GOTO 3350
3415 COLOR 2:PLOT X,Y:IF SQ=0 THEN COL
OR 0:PLOT 0,17:PLOT 0,18:GOTO 3460
3418 X1=INT(38*RND(1)):Y1=INT(18*RND(1
)):MOV=0
3420 LOCATE X1,Y1,W
3430 IF W<>0 THEN 3415
3440 COLOR 1:SETCOLOR 0,4,0:PLOT X1,Y1
3450 FOR LL=1 TO 10:FOR T=1 TO 10:SETC
OLOR 0,0,14:NEXT T:FOR T=1 TO 5:SETCOL
OR 0,0,0:NEXT T:GOSUB 19500:NEXT LL
```

continued on next page

```
3460 GOSUB 2ØØØØ:MOV=MOV+1:IF W=1 THEN
 SQ=SQ-1:FOR L=1 TO 6:SETCOLOR Ø,Ø,14:
GOSUB 191ØØ
3465 IF W=1 THEN COLOR 1:PLOT X,Y:FOR
T=1 TO 2Ø:NEXT T:COLOR 2:PLOT X,Y:FOR
T=1 TO 1Ø:NEXT T:NEXT L:GOTO 3415
347Ø IF W=3 THEN COLOR 2:X=19:Y=9:PLOT
 X,Y:GOSUB 192ØØ:MOV=Ø:GOTO 346Ø
3475 IF MOV=4 THEN COLOR Ø:PLOT X1,Y1:
GOTO 3415
348Ø IF X=Ø THEN 35ØØ
349Ø GOTO 346Ø
35ØØ GRAPHICS 3:COLOR 2:SETCOLOR 1,6,Ø
:SETCOLOR 4,Ø,2:SETCOLOR 2,2,6:GOSUB 1
9ØØØ:C=1:RET=Ø:X=38:OT=Ø
351Ø COLOR 1:SETCOLOR Ø,4,Ø:PLOT X,Y:C
OLOR 3:UP=1:UPP=1:RB=Ø:LB=Ø
352Ø PLOT 9,8:DRAWTO 9,1Ø:IF OT=1 THEN
 354Ø
353Ø PLOT 29,8:DRAWTO 29,1Ø
354Ø IF LB=3 AND RB=2 THEN COLOR Ø:PLO
T 18,19:DRAWTO 2Ø,19
3542 IF LB>=4 THEN LB=Ø
3544 IF RB>=3 THEN RB=Ø
3545 GOSUB 2ØØØØ:OT=1:IF Y=19 AND W=Ø
THEN 37ØØ
355Ø IF W=3 AND X=9 AND UP=1 THEN COLO
R Ø:PLOT X,8:DRAWTO X,1Ø:LB=LB+1:COLOR
 3:UP=Ø:GOSUB 359Ø:GOTO 354Ø
356Ø IF W=3 AND X=29 AND UPP=1 THEN CO
LOR Ø:PLOT X,8:DRAWTO X,1Ø:RB=RB+1:UPP
=Ø:COLOR 3:GOSUB 36ØØ:GOTO 354Ø
357Ø IF W=3 AND X<11 AND UP=Ø THEN COL
OR Ø:GOSUB 359Ø:COLOR 3:UP=1:GOTO 352Ø
358Ø IF W=3 AND X>27 AND UPP=Ø THEN CO
LOR O:GOSUB 36ØØ:COLOR 3:UPP=1:GOTO 35
3Ø
3585 Y=9:GOSUB 192ØØ:GOTO 35ØØ
359Ø PLOT 8,9:DRAWTO 1Ø,9:Y=7:COLOR 1:
PLOT X,Y:GOSUB 193ØØ:RETURN
36ØØ PLOT 28,9:DRAWTO 3Ø,9:COLOR 1:Y=7
:PLOT X,Y:GOSUB 193ØØ:RETURN
37ØØ GRAPHICS 3:COLOR 2:SETCOLOR 1,12,
Ø:SETCOLOR 4,15,4:SETCOLOR 2,9,4:SETCO
LOR Ø,3,Ø:GOSUB 19ØØØ:Y=1:KE=Ø:C=2
371Ø PLOT 8,7:DRAWTO 14,7:DRAWTO 14,13
:DRAWTO 8,13:DRAWTO 8,7
372Ø PLOT 25,7:DRAWTO 31,7:DRAWTO 31,1
3:DRAWTO 25,13:DRAWTO 25,7
373Ø COLOR 3:PLOT 11,1Ø:PLOT 11,4:COLO
R 1:PLOT 28,1Ø:PLOT 28,4
375Ø IF KE=4 THEN 38ØØ
3755 COLOR C:PLOT X,Y:GOSUB 2ØØØØ:IF W
=3 THEN C=3:KE=KE+1:GOTO 3795
376Ø IF C=3 AND W=2 AND Y=13 THEN Y=14
:PLOT 31,1Ø:GOTO 3795
377Ø IF W=1 THEN C=1:KE=KE+1:GOTO 3795
378Ø IF C=1 AND W=2 AND X=8 THEN X=7:P
LOT 14,1Ø:GOTO 3795
3785 IF Y=19 AND W=Ø THEN 4ØØØ
379Ø X=19:Y=1Ø:GOSUB 192ØØ:GOTO 375Ø
3795 GOSUB 191ØØ:GOTO 375Ø
38ØØ PLOT 18,19:DRAWTO 2Ø,19:GOTO 3755
4ØØØ GRAPHICS 3:COLOR 2:SETCOLOR 1,6,4
:SETCOLOR 4,Ø,8:SETCOLOR 2,Ø,Ø:GOSUB 1
9ØØØ:Y=1:C=3:HBR=1:XX=19:YY=9
4Ø1Ø COLOR 3:PLOT 1Ø,8:DRAWTO 1Ø,1Ø:PL
OT 3Ø,8:DRAWTO 3Ø,1Ø:COLOR 1:PLOT XX,Y
Y:DRAWTO XX+2,YY
4Ø2Ø COLOR C:PLOT X,Y:GOSUB 2ØØØØ:IF W
=1 THEN Y=YY-2:COLOR 1:PLOT X,Y:GOSUB
194ØØ:C=1:GOTO 4Ø2Ø
4Ø3Ø IF X=1Ø AND C=1 AND W=3 THEN Y=6:
C=3:GOTO 45ØØ
4Ø4Ø IF X=3Ø AND C=1 AND W=3 THEN Y=6:
C=3:GOTO 46ØØ
4Ø45 IF Y=19 AND W=Ø THEN 9ØØØ
4Ø5Ø GOSUB 194ØØ:X=2Ø:Y=3:C=3:GOTO 4Ø2
Ø
45ØØ IF HBR=1 OR HBR=2 OR HBR=3 OR HBR
=6 OR HBR=9 OR HBR=1Ø THEN CC=Ø:GOTO 4
8ØØ
451Ø CC=Ø:GOTO 47ØØ
46ØØ IF HBR=4 OR HBR=5 OR HBR=7 OR HBR
=8 THEN CC=Ø:GOTO 48ØØ
461Ø CC=Ø:GOTO 47ØØ
47ØØ IF YY<=9 THEN YY=1Ø
47Ø5 COLOR CC:PLOT XX,YY:DRAWTO XX+2,Y
Y:IF CC=Ø THEN CC=1:YY=YY-1:GOTO 47Ø5
471Ø HBR=HBR-1:IF HBR=Ø THEN HBR=1
472Ø GOSUB 191ØØ:GOTO 4Ø2Ø
48ØØ COLOR CC:PLOT XX,YY:DRAWTO XX+2,Y
Y:IF CC=Ø THEN CC=1:YY=YY+1:GOTO 48ØØ
481Ø HBR=HBR+1:GOSUB 191ØØ:GOTO 4Ø2Ø
9ØØØ SEC=((PEEK(18)*256+PEEK(19))*256+
PEEK(2Ø))/6Ø
9Ø1Ø MIN=INT(SEC/6Ø):HR=INT(MIN/6Ø)
1ØØØØ GRAPHICS 18:POSITION 1,1:? #6;"c
ongratulations you":POSITION 1,3:? #6;
"made it!  it took"
1ØØ1Ø POSITION 1,5:? #6;"you:":POSITIO
N 2,7:? #6;"hrs.  min.  sec.":POSITION
 2,8:? #6;HR
1ØØ2Ø POSITION 8,8:? #6;MIN-HR*6Ø:POSI
TION 14,8:? #6;INT(SEC-MIN*6Ø)
1ØØ3Ø POSITION 4,1Ø:? #6;"PRESS START"
1ØØ4Ø IF PEEK(53279)<>6 THEN 1ØØ4Ø
1ØØ5Ø GOTO 5Ø
19ØØØ PLOT Ø,Ø:DRAWTO 39,Ø:DRAWTO 39,1
9:DRAWTO Ø,19:DRAWTO Ø,Ø:RETURN
191ØØ FOR T=1Ø TO 1 STEP -2:SOUND 1,T*
5,1Ø,8:NEXT T:SOUND 1,Ø,Ø,Ø:RETURN
192ØØ FOR T=1 TO 15:SOUND 1,1ØØ,12,8:N
EXT T:SOUND 1,Ø,Ø,Ø:RETURN
193ØØ FOR T=2ØØ TO 2Ø STEP -15:SOUND 1
,T,8,8:NEXT T:SOUND 1,Ø,Ø,Ø:RETURN
194ØØ FOR T=1 TO 3:SOUND 1,2Ø,1Ø,8:NEX
T T:SOUND 1,Ø,Ø,Ø:RETURN
195ØØ FOR T=1 TO 1Ø:SOUND 1,25,12,8:NE
XT T:SOUND 1,Ø,Ø,Ø:RETURN
2ØØØØ TRAP 2ØØØØ:? :? "WHICH DIRECTION
 SHOULD I MOVE";:INPUT D$:? :? "HOW MA
```

```
NY STEPS SHOULD I TAKE";:INPUT S
20040 FOR L=1 TO S:COLOR 0:PLOT X,Y:SO
UND 0,10,0,0
20050 IF D$="U" THEN Y=Y-1
20060 IF D$="D" THEN Y=Y+1
20070 IF D$="R" THEN X=X+1
20078 IF D$="L" THEN X=X-1
20090 LOCATE X,Y,W
20095 IF W<>0 THEN RETURN
20097 IF X=39 OR Y=19 OR Y=0 OR X=0 TH
EN RETURN
30000 COLOR C:PLOT X,Y:SOUND 0,10,0,6:
FOR T=1 TO 10:SOUND 0,0,0,0:NEXT T:FOR
 T=1 TO 15:NEXT T:NEXT L
30010 IF RET=1 THEN RETURN
30020 GOTO 20000
```

TYPO TABLE

Variable checksum = 746160

Line num range			Code	Length
1	-	140	SM	584
160	-	330	BP	512
335	-	367	JW	527
380	-	500	ID	522
510	-	1000	SZ	518
1010	-	1046	WN	556
1047	-	2010	TG	539
2020	-	3000	CW	573
3010	-	3060	VY	502
3065	-	3110	RF	516
3120	-	3150	JH	552
3160	-	3175	DA	546
3200	-	3350	TV	532
3360	-	3450	SR	620
3460	-	3500	LQ	541
3510	-	3550	CS	506
3560	-	3700	EC	661
3710	-	3770	RW	501
3780	-	4010	KQ	513
4020	-	4600	AA	506
4610	-	10000	YN	521
10010	-	19100	TV	502
19200	-	20040	YY	558
20050	-	30020	QA	341

THE 410 HI-REL MOD

Eliminate the most frequent cause of bad loads

by CARL EVANS

In the April 1983 issue of ANTIC, I described a "High-Reliability Modification" (hi-rel mod) for the Atari 410 cassette recorder. Unfortunately, many of you missed that particular issue and I've received numerous requests for reprints. In this month's column I'll go back over it in more detail. My first mention of the hi-rel mod was simply an aside in a column devoted to the 410 recorder's digital-playback circuits. This time I'll concentrate on the mod itself.

A BRIEF DESCRIPTION

First, I'll briefly describe the hi-rel mod and why it works. The 410 and 1010 cassette recorders use two parallel filter circuits that convert the signal recorded on a program cassette into a form the Atari computer can understand. Think of these two filters as a couple of bell-shaped curves, like those shown in Figure 1. Any signal that falls in the band (window) of the low filter is converted into a logical "zero"; a signal that falls into the window of the high filter is converted into a logical "one." Any signal that falls outside the windows of both filters is ignored.

SPURIOUS SIGNALS

Unfortunately, electronics, like life, is not a simple matter of black and white. Due to a number of different causes, spurious signals can get through the filters and cause a bad CLOAD. There is no way to eliminate all of these bad signals, but a careful study of the Atari recorder's digital-playback circuits can provide us with a way to eliminate the most frequent cause of bad loads.

Bad loads are often caused by what might be called "cross-talk" between the two filters. Look again at the curves in Figure 1. You'll notice that they overlap somewhat. As a result, it's possible for the circuit to receive a valid signal but to assign the wrong value to it. Thus, what should have been a zero becomes a one or *vice versa*. The result of this mixup is a bad load.

POSSIBLE SOLUTIONS

Two possible solutions immediately come to mind. First, we could add a new circuit that would blank out the region of overlap. This solution involves the addition of a "notch-filter" to the circuit. Such a circuit modification would be extremely effective, but it would also require the use of an oscilloscope to properly tune the notch-filter. There has to be an easier way than this to achieve our goal. Let's look at the second solution.

The second solution is to narrow the skirts of the two filters so that the region of overlap is minimized. This can be accomplished by replacing a single resistor in the feedback loop of each filter. All that's needed is a soldering iron and a screwdriver. This seems to be the best solution we can come up with without spending a small fortune on an oscilloscope.

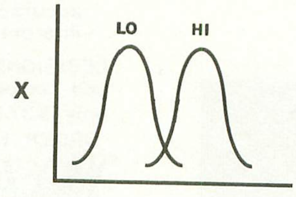


Figure 1

REPLACING THE RESISTORS

First, let's take a look at the two resistors we need to replace. They're highlighted in Figure 2. One of them is a 240K-ohm resistor; the other is a 330K-ohm resistor. Their power ratings are not critical.

Please keep in mind that Figure 2 is a photograph of my recorder — your recorder may have a very different layout. The resistors in my 410 are ten-percent resistors, but some of the newer recorders contain five-percent resistors instead. To significantly narrow the skirts of the bell curves, we need to use one- or two-percent resistors.

Some of you may not know how to "read" a resistor. Look at Figure 3. It shows a side view of a typical resistor. Notice the bands on its side. These are color-coded according to an industry standard that is used to label every resistor. The first band identifies the first digit of a two-digit value. The second band identifies the second digit. The third band shows how many zeros should be added at the end of the two-digit value. For example, on a 24000 (240K) resistor a red band (2) is followed

Carl Evans is a widely published author in various technical and home computer magazines. He is also the author of a best selling book, ATARI BASIC BETTER AND FASTER.

by a yellow band (4), which is followed by another yellow band (X10000). Because of this standardized coding system, our two resistors can be identified by the following patterns:

240K . . RED/YELLOW/YELLOW
330K . . ORANGE/ORANGE/YELLOW

The fourth color band on a resistor is used to show the "tolerance" of the resistor. The specific codes are:

GOLD	5%
SILVER	10%
NONE	20%

The color codes for the first two bands are:

BLACK	0
BROWN	1
RED	2
ORANGE	3
YELLOW	4
GREEN	5
BLUE	6
VIOLET	7
GRAY	8
WHITE	9

The color codes for the third band are:

BLACK	X1
BROWN	X10
RED	X100
ORANGE	X1000
YELLOW	X10000
GREEN	X100000
BLUE	X1000000
VIOLET	X10000000
GRAY	X100000000
WHITE	X1000000000

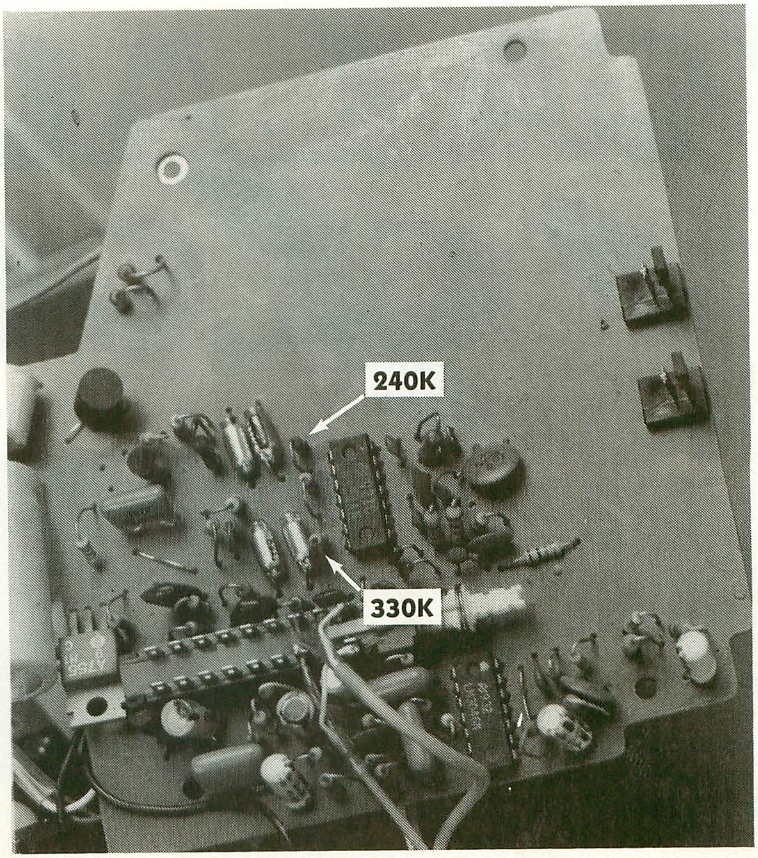


Figure 2
Photo of 410 circuit board

WHAT IS A ONE-PERCENT RESISTOR

Now I'll explain what I mean by a one-percent resistor — the answer is not what you might think. The easiest way to explain the term is to use an example. Let's take the case of a typical 240K-ohm resistor. When these resistors are first tested, the manufacturer's measuring equipment is set to 240K-ohms, plus or minus one percent. Any resistors whose resistance falls between 237.6K and 242.4K pass this test and are labeled "one-percent" resistors. Those that flunk this test are tested with equipment set at 240K plus or minus two, five, ten, or even twenty percent, depending on how many of them fail at each level.

In other words, a ten-percent resistor is a resistor whose resistance is somewhere between 216K and 246K (in our 240K-ohm example). In this case, regardless of a resistor's exact resistance, it is called a 240K resistor. Any given resistor's resistance normally does not vary by more than a fraction of a percent of its stated value; nonetheless, it is rated on the basis of its performance in the screening test.

HOW TO GET THEM

The only real difference between what a vendor calls a one-percent and a ten-percent resistor is price. One way to obtain a one-percent resistor is to buy a batch of resistors and measure their actual resistance with an ohm-meter. If you want to try this, buy cheap resistors. They don't go through extensive screening tests like those I've just mentioned, so their values will vary considerably.



Figure 3
Resistor Color Code Bonding

Another way to get one-percent resistors for the hi-rel mod is to call any good electronics supply house and ask for resistors with the following part numbers: RN-55-D-2433-F (240K ohms) and RN-55-D-3323-F (330K ohms). These are the standard industry part numbers of 1/10th-watt, precision metal-film resistors. You'll be safe with 1/10th-watt resistors, because power dissipation in the part of the playback circuit we're concerned with is less than 1/1000th of a watt.

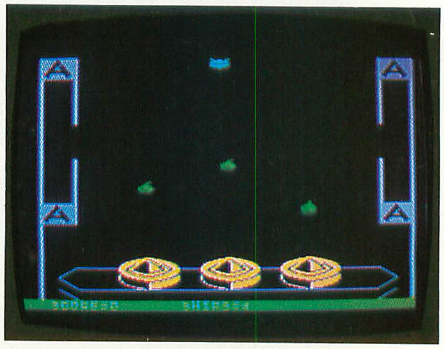
By the way, I was able to locate these metal-film resistors at a local electronics supply store (they were 38 cents each with a minimum purchase of 25) by making two phone calls in the space of about ten minutes. You may not be able to locate them quite that fast, by you should have no difficulty in finding them.

GALACTIC GLOOP

by GUY AITCHISON

Requires 16K cassette, 24K disk

In 1996, two years after America built its first flying saucers, the fiendishly clever warrior race from Cygnus X-1 created its own new method of warfare. Their weapons: thick drops of slime contaminated with explosive space debris. The Cygnians have decided to test their new Galactic Gloop at your moonbase, and it's up to you to defend it.



After typing in the game, run TYPO to check for accuracy and SAVE a backup copy for safekeeping. When you first RUN the game, your saucer (the first of your allotment of five) will appear at the center screen after a brief pause. Beneath you, three drops of gloop are falling; you're carrying one space mine. To stop the dangerous gloop, position your saucer beneath a falling drop with your joystick and press the fire button to deposit a space mine. The gloop explodes on impact with the mine.

To pick up another mine, maneuver into one of the four mine shacks marked with the ANTIC logo. Proceed carefully. If you hit any walls, the crash will be fatal. When you pick up a mine, your saucer turns blue. When you deposit a mine, you hear a "WHOLP" sound. You can carry only one mine at a time, but there is an unlimited supply in the shacks. The game ends when you lose all of your saucers, or if all three ground bases are wiped out. If you hold out long enough, however, The Cygnians will give up, and you'll win.

The first wave of gloop comes three drops at a time; each of these is worth 20 points. In succeeding waves, more drops fall faster, but their point value also increases.

If you lose, the moon blows up and READY appears on the screen. To play again, type RUN and press [RETURN]. Remember, the galaxy is yours to defend.

NOTE: VERY IMPORTANT!!! To run this program in a DOS system (32K or more RAM), you must type in the following *before* LOADing the program:

POKE 743,114:POKE 744,46
[RETURN]
NEW [RETURN]

```
1 REM ** GALACTIC GLOOP **
2 REM * BY GUY AITCHISON *
3 REM *  ANTIC MAGAZINE   *
4 REM *  FEBUARY, 1984    *
5 REM *******************
7 DIM A$(20),XX(30),YY(30):RESTORE 999
:GOSUB 2000
8 XX(1)=6:XX(2)=10:XX(3)=14:XX(4)=8:XX
(5)=12:YY(1)=3:YY(2)=1:YY(3)=5:YY(4)=2
:YY(5)=4:S=1:WA=1:DIF1=3
9 GRAPHICS 17:POSITION 4,8:? #6;"INITI
ALIZING":FOR I=1 TO 400:NEXT I
10 POKE 106,PEEK(106)-5:GRAPHICS 17:DL
=(PEEK(106)+1)*256:POKE 756,DL/256+2:P
OKE 710,180
11 FOR I=128 TO 500:POKE DL+I,PEEK(573
44+I):NEXT I
12 FOR I=1 TO 27:READ O:P=(O+64)*8:FOR
 A=0 TO 7:READ X:POKE P+DL+A,X:NEXT A:
NEXT I
13 DM=PEEK(560)+PEEK(561)*256+27:POKE
DM,2:POSITION 2,22:? #6;" SCORE=0
 SHIPS=5"
22 RESTORE 800:FOR I=0 TO 22:READ A$:P
OSITION 1,I:? #6;A$:NEXT I
23 POKE 711,102:POKE 709,56
30 POKE 559,62:POKE 53277,3:POKE 54279
,40:POKE 704,152:PM=11264:POKE 1777,12
4
31 POKE 1776,124:RESTORE 1008:FOR I=15
36 TO 1656:READ X:POKE I,X:NEXT I:FOR
I=1660 TO 1673:READ X:POKE I,X
32 NEXT I:A=USR(1660):POKE 1776,101:FO
R I=0 TO 7:READ X:POKE I+PM+100,X:NEXT
 I:POKE 53248,124
48 POKE 54286,0:POKE 548,0:POKE 549,6:
POKE 54286,64:RESTORE 3000:GOTO 400
100 IF Q<3 OR Q>17 OR W>15 THEN FF=0:R
ETURN
```

continued on page 80

If you could live on half your income, just think what you could do with the other half!

Lifestyle Budgeting™

the first budget planning system that considers **all** your needs.

Others have done it ... so can you.

Dr. Harper Roehm, the author of ***Spending Less and Enjoying It More*** (the McGraw-Hill book that provides the basis for this system), designed **Lifestyle Budgeting** when his income was cut in half due to a career change from corporate auditor to university professor. As a result, his family has lived comfortably, fulfilling their "wants" as well as their "needs" for over 20 years.

We show you "why" as well as "how."

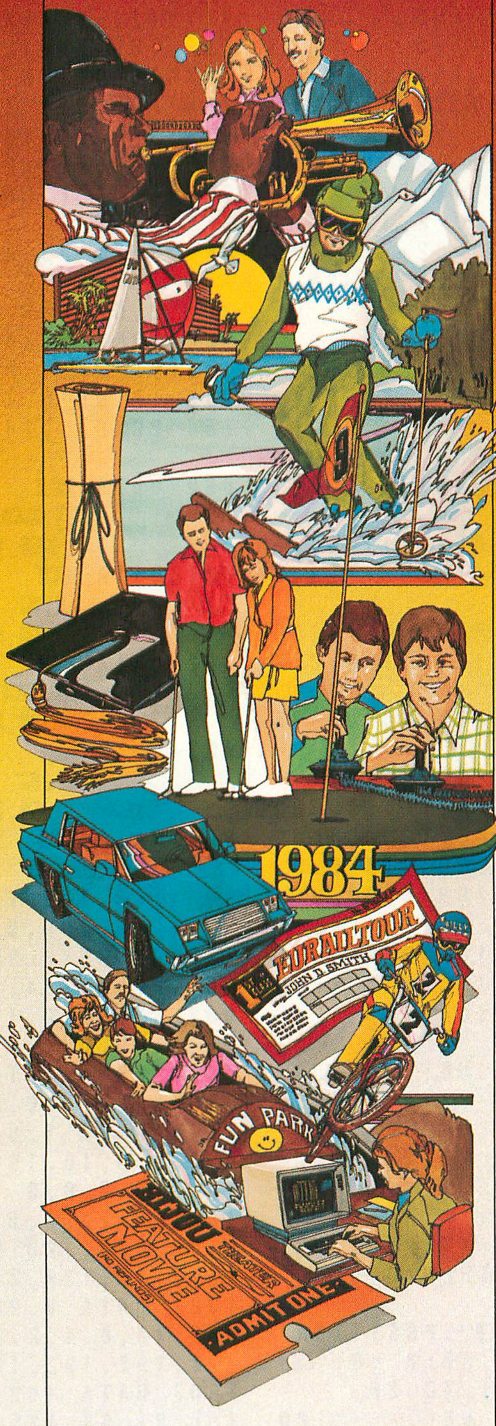
Lifestyle Budgeting is the **first complete budgeting package** to include an easy-to-understand book explaining the behavioral aspects of successful budgeting as well as software to handle the mechanics. It takes a practical approach, showing you not only how you're spending your money, but why ... so you can identify your true priorities and plan for them.

Planning is the key.

Lifestyle Budgeting's software is a forecasting and modeling tool, not a checkbook balancer. It will track your spending patterns, identifying where you're wasting valuable funds. Using that information, and your family's priorities, it will show you specifically how to project and plan future costs.

Little time, and no accounting experience required.

Unlike most budgeting systems, **Lifestyle Budgeting** does not require detailed expense records. It will only take a couple of nights to set up and then only one or two hours a month to monitor. And ... a big plus ... **Lifestyle Budgeting** is written for you, not your accountant.



Use Lifestyle Budgeting on your personal computer.

Since **Lifestyle Budgeting** is something everyone needs, versions are available for IBM, Apple, Commodore, Atari and Coleco personal computers.

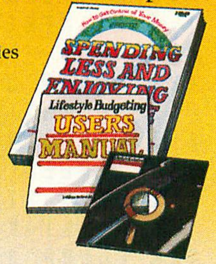
Get control of your money today!

Lifestyle Budgeting will be available through your local computer retailer starting in February ... but **why wait?** You can order the complete package today.

Order now!
Call 1-800-547-1565.
In Ohio Call 1-513-435-2335.

Dealer inquiries welcomed.

Yes! I'd like to know how to set up a liveable budget.



Please send me ______ copies of **Lifestyle Budgeting** (including the McGraw-Hill book ***Spending Less and Enjoying It More,*** and the accompanying software and manual) at $49.95 each plus $3 for shipping and handling. (In Ohio add 6% sales tax.)

Name ______

Address ______

City ______ State ______ Zip ______

Type computer ______ Model ______

Check or money order enclosed ☐

VISA or MasterCard No. ______

Expiration Date ______ Bank No. (MasterCard) ______

Signature ______

Available on diskette only. Allow 4-6 weeks for delivery.

Return coupon and payment to:

culverin Corporation

Culverin Corporation, Lifestyle Budgeting
P.O. Box 503, Centerville, OH 45459.

GALACTIC GLOOP continued from page 78

```
101 POKE 704,102:COLOR 25:PLOT Q,W+2:FOR O=0 TO 15:SOUND 3,10,0,O:NEXT O:SOUND 3,0,0,0:RETURN
202 DM=PEEK(560)+PEEK(561)*256+28:POKE DM,2:FOR I=128 TO 500:POKE DL+I,PEEK(57344+I):NEXT I
400 LL=LL+S:IF LL=1 THEN C1=161:C2=32:FOR I=1 TO DIF1:YY(I)=YY(I)+1:NEXT I
403 IF LL=2 THEN C1=162:C2=163
404 IF LL=3 THEN C1=164:C2=165
405 IF LL=4 THEN C1=166:C2=167:LL=0
410 FOR I=1 TO DIF1:X=XX(I):Y=YY(I):LOCATE X,Y+2,C3
411 READ TU:IF TU=0 THEN TU=193:RESTORE 3000
412 SOUND 1,0,0,0:SOUND 1,TU/2,12,2
418 COLOR C1:PLOT X,Y:COLOR C2:PLOT X,Y+1:COLOR 32:PLOT X,Y-1:IF LL=1 THEN COLOR 32:PLOT X,Y-1
419 IF C3=25 THEN GOSUB 450
420 IF C3>12 AND C3<16 THEN GOSUB 500
422 SETCOLOR 1,I,8:IF YY(I)>17 THEN GOSUB 490
424 IF Q<2 AND W<3 OR Q<2 AND W>11 AND W<13 OR Q>18 AND W<3 OR Q>18 AND W>11 AND W<13 THEN GOSUB 600
425 Q=(PEEK(1777)/8)-6:W=(PEEK(1776)/8)-4:IF STRIG(0)=0 AND FF=0 THEN FF=1:GOSUB 100
427 B=B+0.5:POKE DL+712,0:POKE DL+719,0:IF B=INT(B) THEN POKE DL+712,129:POKE DL+719,129
429 Z=PEEK(53252):IF Z>1 THEN GOSUB 750:POKE 53278,0
438 XX(I)=X:YY(I)=Y:NEXT I:GOTO 400
450 COLOR 25:PLOT X,Y:PLOT X,Y+1:SOUND 0,100,0,14:FOR O=1 TO 10:NEXT O:COLOR 32:PLOT X,Y:PLOT X,Y+1:SOUND 0,0,0,0
451 SC=SC+(10*WA)+10:POSITION 9,22:? #6;SC
452 PLOT X,Y+2:XX(I)=INT(11*RND(0))+5:YY(I)=1:X=XX(I):Y=YY(I):HIT=HIT+1:IF HIT>10+(WA*2) THEN HIT=0:GOSUB 700
459 POP :GOTO 400
490 POSITION XX(I),YY(I):? #6;" ":POSITION XX(I),YY(I)+1:? #6;"[graphics]":YY(I)=1:Y=1
491 XX(I)=INT(13*RND(0))+4:X=XX(I):SC=SC-50:IF SC<0 THEN SC=0
492 POSITION 9,22:? #6;SC;" ":RETURN
500 COLOR 32:PLOT XX(I),YY(I):PLOT XX(I),YY(I)+1:D=9:IF XX(I)<9 THEN D=5
501 IF XX(I)>12 THEN D=13
502 POSITION D,18:? #6;"[graphics]":POSITION D,19:? #6;"[graphics]":POSITION D,20:? #6;"[graphics]":SOUND 0,100,0,14:FOR O=1 TO 20
505 NEXT O:POSITION D,18:? #6;"   ":POSITION D,19:? #6;"[graphics]":POSITION D,20:? #6;"   ":FOR O=15 TO 0 STEP -0.25
507 SOUND 0,100,0,O:NEXT O:SOUND 0,0,0,0:C3=32:DB=DB+1:IF DB=3 THEN 900
509 RETURN
600 IF FF=0 THEN RETURN
601 FF=0:POKE 704,152:RETURN
700 DIF1=DIF1+1:WA=WA+1:SOUND 1,0,0,0:IF WA>15 THEN WA=15
701 SETCOLOR 2,WA,4:POKE 77,0:RESTORE 1100:FOR I=1 TO DIF1*2:READ X,Y:XX(I)=X:YY(I)=Y:NEXT I
702 FOR YH=1 TO 17:COLOR 32:PLOT 3,YH:DRAWTO 17,YH:NEXT YH:RESTORE 3000
705 FOR H=1 TO 5:FOR HH=100 TO 45 STEP -1:SOUND 0,HH,10,10:NEXT HH:NEXT H:SOUND 0,0,0,0:POP :GOTO 400
750 A=USR(1660):POKE 54286,0:POKE 53248,0:RESTORE 1016:FOR O=0 TO 7:READ A:POKE O+PM+33,A:NEXT O
751 FOR O=15 TO 0 STEP -0.25:SOUND 3,130,12,O:NEXT O:POKE 1776,34:POKE 1777,124:POKE 54286,64:SOUND 3,0,0,0
752 DF=DF+1:POSITION 2,23:? #6;5-DF
753 IF DF=5 THEN 900
754 POKE 53248,124:RESTORE 3000:RETURN
800 DATA [graphics]
801 DATA [graphics]
802 DATA [graphics]
803 DATA [graphics]
804 DATA [graphics]
900 FOR I=0 TO 3:SOUND I,0,0,0:NEXT I:SOUND 0,100,0,14:FOR I=21 TO 0 STEP -1:COLOR 25:PLOT 0,I:DRAWTO 19,I:NEXT I
901 POKE 704,0
902 SOUND 0,110,0,14:FOR I=21 TO 0 STEP -1:COLOR 32:PLOT 0,I:DRAWTO 19,I:NEXT I:FOR I=1 TO 200:NEXT I:X=USR(58484)
999 DATA 0,0,0,0,0,0,0,0,0
1000 DATA 1,8,16,48,120,252,188,204,120,2,0,0,16,16,48,120,220,188,3,220,120,0,0,0,0,0,0
1001 DATA 4,0,0,0,0,32,16,48,120,5,204,188,252,120,0,0,0,0,6,0,0,0,0,0,0,16,16,7,48,120,220,188,220,120,0,0
1002 DATA 8,255,0,0,0,0,0,0,0,9,128,64,32,16,8,4,2,1,10,1,2,4,8,16,32,64,128,11,192,192,192,192,192,192,192
1003 DATA 192,12,3,3,3,3,3,3,3,3,13,0,1,7,31,63,125,123,246,14,60,255,231,219,141,14,15,15
```

```
1004 DATA 15,0,128,224,248,252,190,94,
111,16,236,236,247,123,125,191,159,135
,17,15,15,12,195,255,60,195,255
1005 DATA 18,183,55,239,222,62,253,251
,231,19,129,64,64,32,24,6,1,0,20,255,6
0,3,15,15,15,207,40
1006 DATA 21,159,126,254,252,248,224,1
28,0,23,255,254,254,252,252,248,249,24
1,24,255,255,255,127,127,191,63,31,25
1007 DATA 129,102,90,60,60,90,102,129,
26,243,224,224,199,207,239,255,255,33,
159,15,15,199,231,239,255,255
1008 DATA 173,120,2,41,1,208,3,32,43,6
,173,120,2,41,2,208,3,32,67,6,173,120
1009 DATA 2,41,4,208,3,32,91,6,173,120
,2,41,8,208,3,32,106,6,76,98,228,160,8
1010 DATA 174,240,6,202,224,33,144,13,
142,240,6,189,0,44,157,255,43,232,136
1011 DATA 16,246,96,160,8,174,240,6,23
2,224,218,176,245,142,240,6,189,5,44
1012 DATA 157,6,44,202,136,16,246,96,1
74,241,6,202,224,48,144,223,142,241,6
1013 DATA 142,0,208,96,174,241,6,232,2
24,201,176,208,142,241,6,142,0,208,96
1015 DATA 162,255,169,0,157,0,44,202,2
24,0,208,248,104,96
1016 DATA 129,90,60,255,0,255,126,24
1100 DATA 3,1,6,2,9,3,12,4,15,5,4,6,7,
5,10,4,13,3,16,2,3,3,6,4,9,5,12,6,15,7
,4,8,7,7,10,8,13,5,16,4
1101 DATA 3,5,6,6,9,7,12,8,15,9,4,10,7
,9,10,10,13,7,16,6
2000 GRAPHICS 17
2001 POSITION 3,2:? #6;"galactic gloop
":POSITION 2,10:? #6;"by guy aitchison
":POSITION 2,18:? #6;" press any key"
2015 IF PEEK(764)<>255 THEN RETURN
2018 FOR I=1 TO 5:NEXT I:GOTO 2015
3000 DATA 193,126,96,193,126,96,193,19
3,193,126,96,193,126,96,193,193,182,12
1,91,182,121,91,182,182
3001 DATA 193,126,96,193,126,96,193,0
```

TYPO TABLE

Variable checksum = 496012

Line num range			Code	Length
1	-	9	JZ	518
10	-	30	LJ	580
31	-	202	KP	576
400	-	420	GA	530
422	-	450	BD	639
451	-	500	WB	527
501	-	700	TS	561
701	-	751	HY	567
752	-	803	EP	533
804	-	1001	YP	576
1002	-	1007	VB	593
1008	-	1016	ZR	511
1100	-	3001	IJ	495

CP/M ON THE ATARI!

HOW? WITH THE INCREDIBLE

ATR8000

The ATR8000 is a 4 MHz, Z80, 64k RAM, CP/M microcomputer that is also the complete ATARI interface. Besides bringing CP/M to the ATARI home computer, the ATR8000 also enhances the ATARI's operation. The ATR8000 has ports for running a serial or parallel printer for ATARI DOS, complete with a built-in 48k buffer. And, the ATR8000 enables the use of standard 5¼" and 8" drives for ATARI DOS!

The ATR8000-ATARI system runs these DOSes:

ATARI DOS

This is the operating system of ATARI computers. The ATR8000 runs this DOS from any standard disk drive or from an ATARI 810 disk drive.

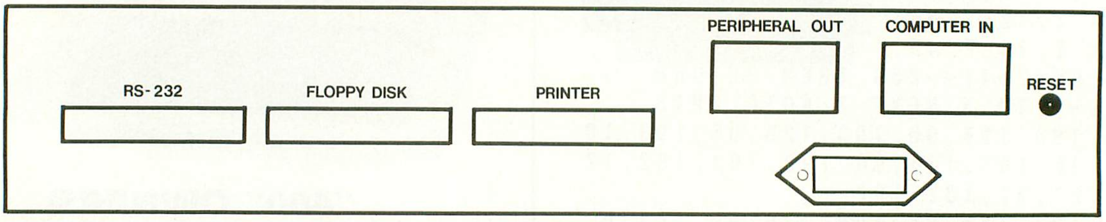
MYDOS

This multi-density DOS is an ATARI compatible DOS. In single density, it runs on an ATARI 810 or a standard drive. In double density, MYDOS runs on standard drives, single and double-sided, 5¼" and 8"!

CP/M

This popular operating system comes with the ATR8000. It is a double density DOS that runs on standard drives. The ATR8000 reads CP/M disks from many other microcomputers, too!

What do the ports of the ATR8000 do?



RS-232: Runs a serial printer or a modem in ATARI DOS and CP/M operation. Includes an automatic 48k printer buffer in ATARI operation. Software includes modem programs.

Floppy Disk: Runs both 5¼" and 8" standard disk drives. Runs up to four drives that are any mixture of size, density (single, double or quad) and type (single-sided or double-sided).

Printer: This standard parallel port runs a parallel printer in ATARI and CP/M operation. Includes an automatic 48k printer buffer in ATARI operation.

Peripheral Out: Interfaces ATARI peripherals to the ATR8000 for use in ATARI operation. Peripherals, like an ATARI 810 drive, are connected with an ATARI Daisy Chain Cable.

Computer In: The ATARI home computer is connected to the ATR8000 here. For CP/M operation, it is also possible to connect an RS-232 terminal here.

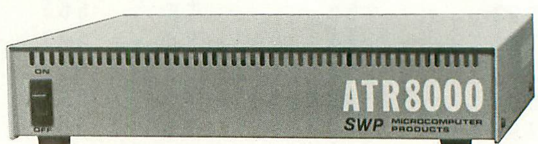
Trademarks: ATARI, ATARI DOS, ATARI 810, ATARI, INC. ATR8000, SWP, Inc. CP/M, Digital Research, Inc. Z80, Zilog.

The ATR8000 is a product of:

SWP Inc
MICROCOMPUTER PRODUCTS

For further information contact your local dealer or SWP at:

2500 E. Randol Mill Road
Suite 125
Arlington, TX 76011
817-861-0421 or
metro 469-1181



64k ATR8000 w/ CP/M $499.95

NIGHTMARE DETOUR

Saved by bank selection

by DAVID and SANDY SMALL

You'll recall from the first three episodes of our "Nightmare Mission" series that we were recently contracted to produce a game for the Atari in roughly one month's time. This was our "nightmare mission." One month is far less than the usual six months allocated for a game's development, and to meet this tight deadline we had to resort to the use of a number of very sophisticated tools and techniques.

In the past three Systems Guide columns we've discussed the equipment and processes we used during the course of this project to get extraordinarily fast results. Our intent is to save the reader time and money by presenting the most efficient means possible to develop Atari programs.

In "Nightmare Mission" (ANTIC, October 1983), we covered the selection of an assembler for the project. We ended up using the MAC/65, which is the speediest and most powerful assembler we've been able to get our hands on. It saved us a great deal of time during assemblies.

In "Mission Accomplished" (ANTIC, November 1983), we discussed the unique debugger we used for the project, the Bit3 board with the ASM/ED cartridge. This provided us with a separate debugging screen in addition to the normal debug printout, and greatly increased the speed of the debugging process.

Finally, in "Mission Redux" (December 1983), we compared the disk drives that are currently available to the software developer and noted the ways in which a fast disk drive can increase your productivity.

This month, after a one-month hiatus, we'll explore the basics of "bank selection," the newest and "hottest" technique for programming Atari cartridges. Only very recently have bank-selected cartridges become available; in this column, we'll show why they're such powerful new tools for the general Atari user and the software developer alike.

David and Sandy Small are professional programmers who work extensively with Atari computers and Atari-compatible peripherals and software to produce commercial software for the Atari. In Systems Guide, they share discoveries, insights, experiences and secrets of professional programming that should be of interest to others who are at or near their level of practice.

CARTRIDGE DESIGN

Let's start with a quick overview of the development of the Atari cartridge. The first cartridges had very little memory. **Basketball** and **TicTacToe**, for instance, each contain only one 4K memory chip. **Star Raiders** and the BASIC cartridge each include two 4K chips, for a grand total of only 8K of memory apiece. When these chips were first offered, prices were very high; remember, Star Raiders used to retail for $60!

Gradually, though, prices fell and chips with more memory were introduced. It isn't uncommon nowadays to find two 8K memory chips (for a total of 16K) contained in a single cartridge such as **Donkey Kong**.

However, you won't be able to find cartridges with more than 16K of memory. Why not? Well, when the Atari computer was designed, room for only 16K was allocated electrically for the cartridge slot. As a result, cartridge designers were stuck with "address lines" that make it impossible to hook up more than 16K in the machine. Keep in mind, though, that back in 1978 16K of memory was an expensive proposition. When Atari's designers allocated space for a 16K cartridge, they thought they were "over-designing" the computer.

WHAT IS BANK SELECTION?

How does all of this relate to our game

continued on next page

project? After we'd finished developing the game and after every demand of our contract had been thoroughly covered, we realized that our game, which had to fit into a 16K cartridge, was 24K long!

At this point, we had several choices. We could trim the game, but this would mean that we wouldn't meet the terms of our contract. We could change it to a disk-based game, and again fail to live up to our contract's demands. Or we could develop a new kind of cartridge. We chose the latter option, of course, and came up with a cartridge that incorporated bank selection.

Bank selection is based on a simple concept. Imagine, for example, that you have a circuit board which contains two 16K chips, and that both of them are at the same memory address. At any one time, the Atari only "talks" to one of these chips, and the other is inactive. Sometimes the computer selects one "bank" of memory, sometimes the other. In other words, 32K of memory is present on the circuit board, but only 16K is connected to the Atari at any given time. This is an important point, because 16K is, of course, the maximum amount that the Atari will let you connect.

If, however, you design a program that can run in one 16K piece out of one chip, and can then "switch in" the other 16K chip and run on it, you'll have fooled the Atari into accepting your 32K cartridge.

HOW WE FOOLED THE ATARI

To trick the Atari into thinking that our cartridge was only 16K, we took two 27128 EPROM chips and bank selected one of them. Since the 27128 is a 16K chip, this gave us 32K of memory on our circuit board. Next, we "mapped," or connected our chips to the Atari in the following manner:

— — — — Memory Address $8000

Chip 1: Bank Selected (4K memory space). Only one 4K bank is active at any one time. 4 banks of 4K are available.

— — — — Memory Address $9000

Chip 2: Not Bank Selected (12K memory space). 12K is always active; 4K is not used.

— — — — Memory Address $C000

In other words, our cartridge had 12K of memory in one chip and four 4K banks of memory in the other; this gave us a total of 16K on-line and active at all times.

How did we accomplish this at the circuit level? Actually, Atari provides almost everything you need. Use a 74LS175 chip, which is a "latch." Connect the "clock" input, which controls when the "latch" occurs, to the D5xx select that is made available to cartridges. Then connect two of the data inputs to D0 and D1, the system data lines. Finally, connect Q0 and Q1, the outputs, to A12 and A13 of the 27128 to be bank selected. As a result, when you:

```
LDA  # (0,1,2 or 3)
STA  $D500
```

one of the banks (0, 1, 2, or 3) will suddenly appear in the memory region from $8000–$9000. The 74LS175 will then be triggered by the D5xx memory-access strobe, and the data line's value will set the 4K selectors (A12 and A13) of the 27128.

After going through these steps, we had 16K available to us in 4K sections, but only one section was available at a time. We also had 12K that was always available. Thus, our new cartridge provided us with 28K of available space. Since our game program was only 24K long, we had 4K to spare.

When you use this procedure, some miscellaneous fiddling is required to disable one ROM chip when the other one is active, and to handle similar tasks, but this isn't too difficult. Bank selection is the important thing.

BREAKING UP THE PROGRAM

We chose to bank select 4K banks because of the nature of our program. It broke pretty easily into four sections of 4K each, and these were placed under

the control of the 12K "master" section.

Our game program included four display screens: The introduction, the "ship" page, the "flying" page, and the "landing" page. These corresponded nicely with our four banks. Whenever we needed to go to a new section, we simply flipped in a new 4K section of the 16K ROM. Most of the data in our game was tied in with scrolling terrain maps, so we put the maps into the banks; after all, terrain maps are not used simultaneously. All of the subroutines and drivers, on the other hand, stayed in the 12K of non-switched memory. This is an excellent technique to keep in mind if your application runs over 16K, and if there's a segment of your program that doesn't need to be available at all times.

You should note, though, that you can bank select just as easily on 8K banks if your application "breaks" better along those lines. For instance, 8K is the size of the display memory in ANTIC modes E and F (Graphics Mode 8); by using 8K banks, you can hide a complete display and flip it back in with one memory write.

64K CARTRIDGES!

In addition, the 27256 (32K) EPROM chip is now available. This means that you can now easily make a 64K cartridge. Only 16K can be used at any one time, but because bank selection is such a fast process, anything you need to access in the cartridge is almost always instantly available.

Imagine a 64K cartridge! This is a huge amount of memory for a computer game, even a game with a lot of text. For instance, imagine the "Zork" games, from Infocom, running from a cartridge with no disk access to slow things down: The computer would practically give you your answer before you asked the question! Chip prices are coming down, and a number of Atari users who don't own disk drives want software they can use, so in all probability the days of bank-selected cartridges are here to stay.

BURNING THE 27128 EPROM

Incidentally, you "burn" the 27128 EPROM (28-pin) using a 2732A (24-pin) EPROM burner; you hand-select which 4K bank you burn.

Be careful of pin 20, which you used to apply programming voltage to; if you do this to a 27128 — despite claims of "pin-to-pin compatibility" — you'll fry the chip. They're $50 each, so this was a tough lesson for us to learn. It turns out that "compatibility" with the 2732 only applies to *reading* the chip.

PIRATE-PROOF BANK SELECTION

I recently picked up a demo copy of OSS's new ACTION! language, which is cartridge based. It uses a "proprietary memory system" to "pack 24,000 bytes of code into only 8K of memory space."

Now you know how this was done: Bank selection. I opened the ACTION! cartridge to see exactly how they did it, and there was my old friend, the 74LS175. It was connected to two 2764 (8K) chips, and was busily bank selecting them. These two ROM chips occupy the same memory space and flip back and forth between themselves periodically, sharing 16K of code in one 8K space.

This arrangement has a unique advantage: It is extremely difficult to make a pirate copy of this cartridge for a disk-based format.

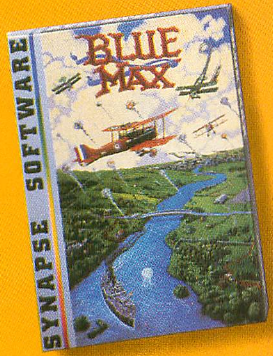
Nearly every other cartridge on the market has been copied to disk format and then distributed widely. The memory space that the cartridge used to occupy is loaded by the disk drive, and the system is fooled into thinking that a cartridge is present. Not so with ACTION! When the bank-selection technique has been used, you can't get to the whole cartridge at once; furthermore, two separate memory images occupy the same memory space. As a result, there is no practical way to simulate this configuration in RAM without relocating the code, which is an extremely difficult task.

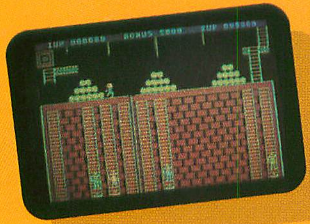
As a software developer, you should keep in mind that bank selection is very effective against pirating. Even if you don't *need* to bank select, because your application fits within 16K, you might want to use it because of the protection it offers. If you use bank selection, your program will not run unless your cartridge is "really" there. That's about the best insurance you're likely to find in this business.

SYNAPSE EXCITEMENT

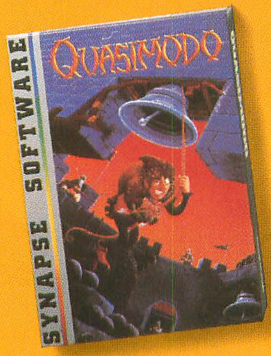


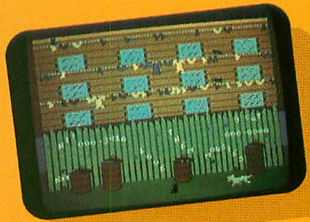
On patrol
Out of the sun comes your RAF biplane, loaded down with a deadly cargo of bombs and bullets. But watch out for the anti-aircraft guns and the enemy fighters—a hit could mean a tricky landing for repairs and ammo. **BLUE MAX.***



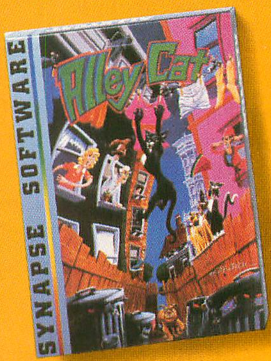


Stop the ringing!
Quasimodo knows who stole the crown jewels. He even knows where they are, but the soldiers just won't leave him alone. Help Quasi dodge the guards and return the jewels to their rightful owner. **QUASIMODO.***



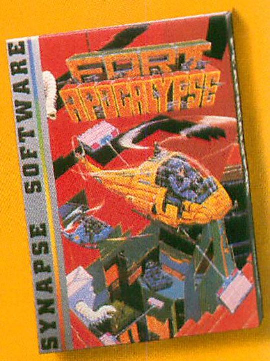


Catastrophic!
Freddy the Cat is busy all day long, getting into trouble every time he jumps in a window. If he's not dodging bottles, bones and shoes he's chasing mice or avoiding that big bully Bowser von Spike! **ALLEY CAT.***





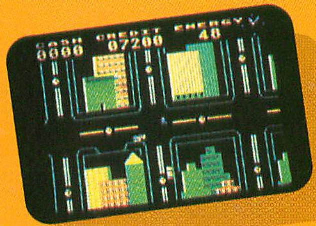
Take the controls
Your helicopter mission—capture vital fuel and weapons, free the enslaved masses, and finally destroy the fortress itself. Will you triumph or be crushed by the fiendish Kraalthanlords? **FORT APOCALYPSE.***



FOR YOUR ATARI!

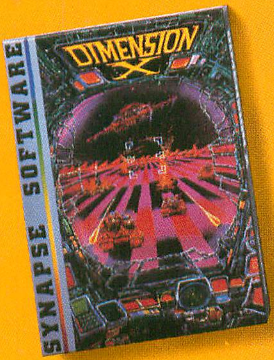




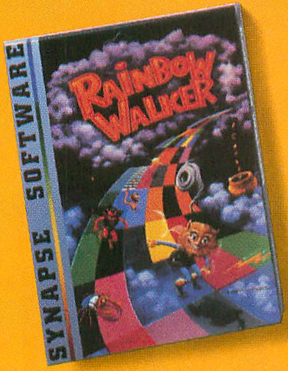




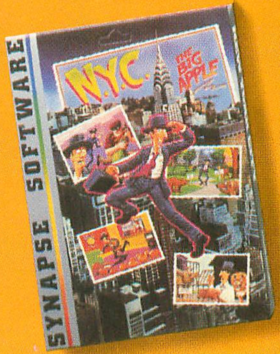
Surrounded!
There's nothing like a Rigillian invasion fleet to spoil your day! You just have time to activate the shield envelope and weapons systems before the enemy ships fill the scanner screen. Good luck—you'll need it! **DIMENSION X.***



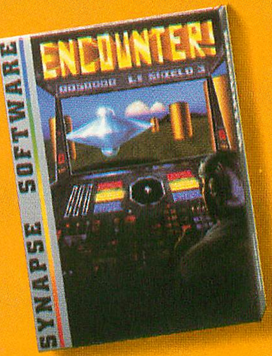
Color the world
Someone stole all the colors of the rainbow. It's up to Cedrick to put on his magic shoes and hop from square to square putting back the color. But watch out for the Borbuls, the Devils, the Gruebirds and the big holes in the rainbow! **RAINBOW WALKER.***



Big Apple bites back
A typical Midwestern tourist takes his first big bite out of the Big Apple. Little did you realize just how much trouble you could get into in broad daylight! **NEW YORK CITY.***



Can you survive?
Alone on a distant planet, you stumble into an ancient combat training course that really plays for keeps. Can you survive the deadly drones and sneaky saucers? **ENCOUNTER.***



Write for a FREE catalog and Elite Club information.

Synapse games are also available on disk and cassette for the Commodore, Apple and IBM home computers.

"A 2.84"

Name ______________________

Address ______________________

City ______________________

State ____________ Zip ____________

Computer ______________________

synapse
SOFTWARE

5221 Central Avenue, Richmond, CA 94804 · 415/527-7751

LOOK WHAT WE HAVE FOR YOUR ATARI® HOME COMPUTER

More Disk Drive for your money. In fact, with the Astra 1620, you get **two** superb Disk Drives for the price of one. The Astra 1620 is single or Double Density (software selectable) and completely compatible with Atari Dos or OSA+ Dos. When used as Double Density the Astra 1620 has the same capacity as Four Atari 810® Disk Drives.



(7-7/8'' WIDE × 11-7/8'' DEEP × 5-7/8'' HIGH)

The ASTRA 1620 DOUBLE DENSITY – DOUBLE DISK DRIVE HAS A SUGGESTED LIST PRICE OF $595.00.

ASTRA SYSTEMS

5230 CLARK AVENUE • LAKEWOOD, CA 90712 • PHONE: (213) 804-1475

® ATARI IS A REGISTERED TRADEMARK

DRAWTO FILL

Color within the lines

by JOSEPH TUCKER

Here is a useful and instructive assembly language program that draws areas and fills them with color. I assume that you have the necessary tools: the Assembler Editor cartridge, the *Technical User Notes*, and a book on the instruction set for the 6502 chip.

This program sets up a full screen in Graphics 3, and then fills a box in the upper left portion of the screen. I've included lots of extra stuff for you to experiment with. The program is meant to be illustrative, and does not necessarily reflect good programming practices.

To start the program, we open a channel (IOCB #6) to the screen. Lines 86 and 88 enter the value of a color and put it in the register for Playfield 1. Lines 90 and 96 put the data value of Playfield 1 in ATACHR (see "Appendix H" of the *Technical User Notes*).

Lines 98–108 plot points to DRAWTO. Line 112 sets the command byte for drawing; line 116 lets the handler subroutine execute the command. Lines 118–134 complete the drawing of the box.

ROWCRS and COLCRS should end at the lower left corner of the figure. Then, in lines 136 + 138, we put the current cursor in the upper left corner. In lines 140–150 we set up the fill color, set the fill command byte, and fill the figure using the handler.

Finally, in line 152, the image is held on the screen with an endless loop.

System Requirement: 16K, Assembler Editor

```
02 ;GRAPHICS, DRAWTO AND FILL
04 ;PAGE #'S FROM 1982 TECH. REF. NOTE
S
06 ;PROGRAM BY JT.
08 ;ASSEMBLE AND TYPE "BUG"; "G0602"
10 ;
12 ;SET UP PLAYFIELD COLOR REGISTERS
14 ;pg. 62
16 COLPF0 = $2C4
18 COLPF1 = $2C5
20 COLPF2 = $2C6 ;TEXT WINDOW
22 COLPF3 = $2C7
24 COLPF4 = $2C8 ;BACKGROUND
26 ;SOME COLORS YOU CAN USE pg. 190
28 GOLD = $18
30 REDOR = $34
32 BLUE = $78
34 GREEN = $C4
36 ;
38 ;CURSOR REGISTERS pg. 61,212-225
40 ROWCRS = $54
42 COLCRS = $55
44 NEWROW = $60
46 NEWCOL = $61
48 CRSINH = $02F0
50 OLDROW = $5A
52 OLDCOL = $5B
54 FILDAT = $2FD ;FOR FILL COLOR
56 ATACHR = $2FB ;FOR DRAW COLOR
58  *=$0600 ;STARTING ADRESS
60 DEV .BYTE "S:" :SCREEN
62  LDX #$60 ;IOCB #6
64  LDA #$3 ;OPEN
66  STA $342,X ;ICCOM COMMAND CODE
68  LDA #DEV&$00FF ;MASK OFF HI BYTE
70  STA $344,X ;ICBAL BUFFER ADR. LO B
YTE
72  LDA #DEV/256 ;HI BYTE
74  STA $345,X ;ICBAH
76  LDA #$0C ;READ/WRITE. SPLIT SCREEN
78  STA $34A,X ;ICAX1
80  LDA #3 ;GR. 3
82  STA $34B,X ;ICAX2
84  JSR $E456 ;LET CIOV HANDLER DO IT
86  LDA #REDOR ;RED-ORANGE FOR PF1
88  STA COLPF1
90  LDA #2
92 ;IN GR.3 DATA VALUE FOR PF1 = 2
94 ;SEE TECH. NOTES APPENDIX H, 60-62
, 188-189
96  STA ATACHR
98  LDA #0 ;PLOT AND
0100  STA OLDCOL
0102  LDA #12
```

continued on next page

```
0104   STA OLDROW
0106   STA COLCRS ;DRAWTO POSITIONS
0108   STA ROWCRS
0110   LDX #$60 ;IOCB #6
0112   LDA #$11 ;DRAW COMMAND pg. 60,2
18
0114   STA $342,X ;ICCOM
0116   JSR $E456 ;CIOV
0118   LDA #0 ;DRAW NEW POSITIONS
0120   STA ROWCRS
0122   JSR $E456
0124   LDA #0
0126   STA COLCRS
0128   JSR $E456
0130   LDA #12
0132   STA ROWCRS
0134   JSR $E456
0136   LDA #0
0138   STA ROWCRS
0140   LDA #1 ;PF0 COLOR DATA
0142   STA FILDAT ;FILL WITH PF0 COLOR
0144   LDX #$60
0146   LDA #$12 ;FILL CODE
0148   STA $342,X
0150   JSR $E456
0152 STOP JMP STOP ;HOLD ON SCREEN
0154 ;NOW MAKE YOUR OWN PLAYFIELDS
```


COMPUTER PALACE
Over 1000 Items For Your Atari!
ATARI®
We Know ATARI Computers... Call Us For The Latest Info: (503)683-5361
the Koala Pad
From Koala Tech.
new
The future is here... Input your commands with the touch of a finger on your Koala Pad.
● Super Graphics & Drawing Program Included with Pad.
Retail $100.00
$87.00 Plus Freight
ATARI COMPUTER
DISK DRIVE
DUST COVERS
Made from durable brown vinyl & custom fitted for:
● ATARI 400, 800, 600XL, 1200XL Computers
● ATARI 410 Recorder, 810 Disk Drive
● EPSON MX-80 or FX-80 Printers
$8.95 EACH
Please specify which model
FREE Catalog
With any order, or send $1 (refundable with purchase). Send us your name & address for FREE QUARTERLY FLYERS with SPECIALS!
Over 1000 items for your Atari...
● Mosaic ● Percom
● Broderbund ● APX
● Roklan ● Datasoft
● Synapse ● A I
● On-Line ● Atari
● Hayes ● Epson
● Visicalc ● More!
COMPUTER PALACE
DEALING EXCLUSIVELY IN ATARI
New Products!
New Lower Prices!!!
Use Your Credit Cards!
We honor VISA, American Express, & MasterCard.
24 & 48 Hour Shipping Is Available... CALL!
ATR-8000 EXPANSION UNIT
● Run CP/M on your Atari ● Parallel printer port
● Printer spooler ● Serial port ● DD controller
Plus Freight
16K model. 299.00
64K model (for CP/M). 499.00
2-DDDS Drives (for 64K model). 695.00
Ape-Face
Parallel Printer Interface
$89.50
PINBALL (Much more than a game!) CONSTRUCTION SET
48K Disk
New from Electronic Arts
$35.10
● Design the pinball game of your dreams, or play one of five games on the disk! Share your games with friends!
ALSO FROM ELECTRONIC ARTS...
M.U.L.E (D). 35.10
Hard Hat Mack (D). 31.50
Archon (D). 35.10
Axis Assassin (D). 31.50
Murder on the Zinderneuf (D). 35.10
Worms (D). 31.50
This is one of the Most Versatile data-base programs available for business or home use!
new
Super Mailer PLUS
From ROYAL SOFTWARE 48K Disk only
Much More Than A Mailing List!
Maintain Your Book Library
Organize Your Record Collection
Keep Track Of Your Recipies
Are You A Stamp Collector?
Let Super Mailer Plus keep track of all these and more!
Dealer Inquiries Invited
Our Price $49.95
You can accomplish this and more because of the "User Re-defineable fields"! The program has many other features: Multiple Search—Special Code Field—Single/or/Double Density Operation—Create Sub-files—Auto-delete—Print Labels 1/2/or 3-up—Print File Copy—Merge/Search/Sort—Built-in Official Post Office State Abbreviations—More.
$54.00
The Monkey Wrench II
A cartridge for the right-hand Slot of your Atari 800!
This is a BASIC & Machine-language Programmers Aid for the 800. It plugs into the right slot, and works with Atari BASIC to add 16 new direct-mode commands(Incl. Auto line numbering, Delete lines, Renumber BASIC, Hex-Dec conv., Find string, Move lines, Exchange string, Disk directory from BASIC, Mem. test, More.) Also includes a monitor with 15 commands to interact with the 6502. This has been proven to be a very helpful tool when writing your own programs!
ALL-TIME FAVORITES
Star Raiders(C). 38.20
Eastern Front (D,T). 29.95
Pacman (C). 39.50
Temple of Apshi (D,T). 35.10
Zork I, II, or III (D). . . . . each 35.10
Centipede (C). 39.50
Missle Command (C). 29.70
Shamus (D,T). 31.50
Ghost Hunter (D). 34.95
Ghost Hunter (T). 29.95
TOP-TEN PROGRAMS
Pole Position (C). 44.90
Donkey Kong (C). 44.90
Blue Max (D,T). 31.50
Baseball (D,T). 28.90
Pinball Construction Set (48K, Disk). . . 35.10
Football (D,T). 28.90
Monkey Wrench II (not a game) (C). 54.00
Ultima II (D). 52.50
Archon (D). 35.10
Dig Dug (C). 44.90
FLIGHT SIMULATOR II From Sublogic 48K Disk $44.90
Colorful, scrolling graphics—Full instrumentation
The program includes a World War I "Air-Ace" battle game, too!
ULTIMA III/Exodus
$49.50
EXODUS ULTIMA III
From Origin Systems
Here is the latest in the famous series of fantasy/role-playing adventure games.
HARDWARE
Astra 1620 Disk Drive. 529.00
Micromainframe Disk Drive. 499.00
Rana 1000 Disk Drive. 349.00
Trak A2-D2 Disk Drive. 398.00
Super-5 Printer. 329.00
Panasonic Monitor (Grn/Scrn, w/sound) 199.00
SOFTWARE
Archon (D,T). 35.10
Joust (C). 44.90
Robotron (C). 39.50
Microsoft Basic II (D). 79.10
Diskey Utility (D). 44.90
Quasimodo (D,T). 31.50
Murder on the Zinderneuf (D). 35.10
Blue Max (D,T). 31.50
Odessa Chess (D). 59.50
Pharoah's Pyramid (D). 31.50
Rally Speedway (C). 44.90
Ultima II (D). 52.50
Ultima III/Exodus (D) 49.50
Tax Advantage (D)(Not a game). 59.95
SUPER SPECIALS—UP TO 60% OFF!
Meteor Storm
24K Disk Only From Royal Software
Special $9.95
Special $17.95
K-Razy Cartridge Shootout
From CBS
COMPUTER PALACE
OPEN Monday-Saturday, 9 to 6
2160 W. 11th Avenue Eugene, Oregon 97402
USE YOU CREDIT CARD & CALL TOLL-FREE 1-800-452-8013
You Never Pay Extra To Use You Credit Card At Computer Palace!
Our Toll-Free Number Is For ORDERS ONLY, PLEASE! Call 503-683-5361 for info.
Shipping Note: UPS & Parcel Post, Add $2.00—UPS Blue Label (Air), Add $3.90—Hardware and Paper Supplies are Extra—Call for exact cost on your order. 24 & 48 hour shipping available—Call 503-683-5361 for information.

A CINDERELLA STORY
continued from page 18

scan is completed in roughly one sixtieth of a second. At the end of a scan, the electron beam is turned off and aimed again at the top-left portion of the screen in preparation for a fresh scan. The period during which the electron beam is off is called the vertical blank. Machine-language programmers take advantage of this time to make graphics changes and move players and missiles. Since these changes are made every sixtieth of a second, the resulting animation is smooth and clean. Another use for the VBI period is to play music or produce sounds that are independent of the main program.

During the VBI, the OS performs a number of housekeeping tasks. These include incrementing or decrementing different timers, updating readings from controllers and copying data from one register to another. Two addresses are used to help the processor find the VBI routines. The programmer can change these addresses to force the 6502 to perform a different routine. Of course, one must be careful to direct the 6502 back to the correct place in the VBI routine Otherwise, the 6502 will run out of control, causing the computer to become unresponsive to commands. If this happens, the only solution is to turn the computer off and then back on. The system monitor will then come into play and initialize the computer's memory registers properly. As a result, the beast will once again become domesticated.

INTERRUPTING THE DISPLAY LIST

Remember the display list, the mini-program used by ANTIC to control the screen display? The programmer can modify one of the display list instructions to cause an interrupt (called a display list interrupt or DLI) to occur at a precise scan line on the screen. The 6502 will then go to a certain memory location, get an address, go to that address and execute the program it finds there. Of course, the programmer must have previously loaded his DLI program into memory and placed the program address in the proper place.

At the start of each screen scan line there is a period during which the electron beam is off. This is called the horizontal blank. Any changes made during this time will not be seen until the electron beam is turned back on. Again, this allows neat, clean graphics to be made. The time period of the horizontal blank is much shorter than that of the vertical blank. Thus, only a limited number of changes can be made during a single DLI. But since a DLI can be set at every screen scan line, if desired, this can be a very powerful tool. For example, multiple players can be made to appear on the screen when the program is actually using only one player. At various positions on the screen, DLI's can be used to change a player's horizontal position, or even its image. **Joust** is a good example of this technique. I've provided a useful DLI routine, to be used from BASIC, in the paragraphs below.

PLOTTING COLORS

If you've read the BASIC manuals, you know that Graphics Modes 3, 5 and 7 are four-color modes. One color is used for the background. So, in theory, you can plot independently in three colors. If you are using a text window, however, you will find that things aren't that

Listing 1

```
1 REM *** OS DEMO 1 ****
2 REM * BY FRED PINHO    *
3 REM * ANTIC MAGAZINE   *
4 REM * FEBUARY, 1984    *
5 REM ******************
10 GRAPHICS 3:F=0:? "NORMAL SCREEN(NO DLI).":? "CAN YOU SEE THE LETTERS?";:REM OR GR.5 OR GR.7
20 COLOR 1:PLOT 9,0:DRAWTO 9,19
30 COLOR 2:PLOT 19,0:DRAWTO 19,19
40 COLOR 3:PLOT 29,0:DRAWTO 29,19
50 POKE 752,1:GOSUB 300
60 SETCOLOR 0,5,6:GOSUB 300
65 SETCOLOR 0,2,8:GOSUB 300
70 SETCOLOR 1,2,6:GOSUB 300
80 SETCOLOR 1,2,4:GOSUB 300
90 SETCOLOR 1,12,10:GOSUB 300
100 SETCOLOR 2,12,4:GOSUB 300
110 SETCOLOR 2,12,10:GOSUB 300
120 SETCOLOR 2,9,4:GOSUB 300
130 IF F=1 THEN POKE 54286,64:GOTO 10
135 REM POKE OF 64(NORMAL VALUE) INTO 54286 DISABLES DLI
140 ? CHR$(125):? "SETTING UP DLI!";:F=1:GOSUB 300
150 DL=PEEK(560)+256*PEEK(561)
160 DLI=DL+24:REM FOR GR.5 USE DL+44,FOR GR.7 USE DL+84
170 POKE DLI,136:REM FOR GR.5 POKE DLI,138 AND FOR GR.7 POKE DLI,141
180 RESTORE 190:FOR I=0 TO 15:READ D:POKE 1536+I,D:NEXT I
190 DATA 72,169,202,141,10,212,141,23,208,169,148,141,24,208,104,64
200 POKE 512,0:POKE 513,6:POKE 54286,192
205 REM LOCATIONS 512 & 513 HOLD ADDRESS OF DLI ROUTINE
207 REM POKE INTO 54286 ENABLES DLI
210 ? "  DONE!":? "CAN YOU SEE THE LETTERS?";:GOSUB 300:GOTO 20
300 FOR T=1 TO 500:NEXT T:RETURN
```

TYPO TABLE

Variable checksum = 88021

Line num range	Code	Length
1 - 65	FJ	476
70 - 170	PU	549
180 - 300	QE	356

continued on page 97

PUBLIC DOMAIN SOFTWARE

ANTIC GAMES DISK #1

1. Chicken: a great game from ANTIC V.1, No.1
2. Hangman: the traditional word game
3. Creation: a version of Life
4. Reverse: order of string numbers
5. Monopoly: computerized!
6. Lunar lander: select terrain (stick)
7. Zonex: hidden color patters in grid
8. Clewso: detective adventure, graphics

ANTIC GAMES DISK #2

1. Speed Demon: from ANTIC V.1, No.4
2. Guy' Grid Game
3. Deathstar: from ANTIC V.1, No.2
4. Blackjack: Vegas rules
5. Civil War: a strategic simulation
6. Artillery: firing strategy game
7. Super Wumpus: text adventure

ANTIC GAMES DISK #3

1. Petals
2. Showdown
3. FROG: from ANTIC V.1, No.3
4. Draw: Guy Hurt
5. Plus Zero
6. Collisi, and more

ANTIC GAMES DISK #4

1. Vultures: from Stan Ockers
2. Castle Hexagon: also by Stan Ockers
3. Adventure: The remainder of the disk contains an adventure game program which you can play, or you can use the program to write your own adventure. Instructions included in program.

ANTIC GAMES DISK #5

1. Bats: who else? Stan Ockers, from ANTIC V.1, No.5
2. Stellar Defense: from ANTIC V.1, No.6, slightly improved
3. Yahtzee
4. Mastermind
5. Hamurabi: the classic simulation
6. Slalom: become a master schusser
7. Couch: analyze yourself
8. Aceyducy & more

ANTIC GRAPHICS DEMO #1

1. Spider: from ANTIC V.1, No.3
2. Rainbow
3. Horses
4. ATARI logo
5. Oxygen
6. Spiral
7. Pretty
8. Message and more

ANTIC PHOTO GRAPHICS

Digitized Photos

ANTIC UTILITY DISK #1

1. Doc: program allows you to accompany programs with separate documentation on disk.
2. Microassembler: allows you to create USR routines-assembler, more
3. Assembler-Editor: BASIC, slow but versatile
4. Num: automatic line numbering utility in BASIC
5. Memtest: runs without BASIC cartridge, to test all memory
6. Color: 128 colors at once
7. Printnop: connect parallel printer from jacks 3 & 4

ANTIC UTILITY DISK #2

1. Bubble Sort: From ANTIC V.1, No.4
2. Typo: from ANTIC V.1, No.3
3. Home inventory
4. KEY 6: Cipher coding
5. Renumber
6. Compare: listings for differences
7. SUPER: menu
8. Modem
9. RT clock & more

ANTIC UTILITY DISK #3

1. Disassembler: from ANTIC V.2, No.1
2. Tiny Text: from ANTIC V.1, No.6
3. GTIA text window: from ANTIC V.2, No.1
4. Label: disk label on Epson
5. Set up printer: sets up MX80 for Visicalc
6. Keyboard: tutorial & more

ANTIC GR. & SO. DEMO #1

1. Graphic
2. Draw
3. Rainbow
4. Tune Rite
5. Etch Sketch
6. Baby Pro Sound and more

ANTIC MUSIC DISK #1

Requires Music Composer Cartridge

1. Prelude
2. Joplin
3. In My Life
4. Star Trek
5. Daisy
6. Greensleeves
7. Yellow Submarine, and many more

Antic is pleased to offer a library of Public Domain Software for the Atari computers currently comprised of 12 disks. These programs are not yet available on cassettes. These disks contain unprotected material from the libraries of Atari users' groups from around the country.

The potential buyer should note that these programs are sold *as is*. Their usefulness may depend on your experience with the computer. They may contain programming quirks that require some modification. However, all perform reasonably well. Contents of the disks may vary slightly from the published description due to unforseen circumstances, but each disk is filled to reasonable capacity with useful programs of the kind described. In most cases, there is no documentation except that in the program themselves. An excellent value at $10.00 each, plus $1.50 per order for shipping/handling. Send check or money order (payable to ANTIC Publishing), and disk number(s) to:

Public Domain Software
524 Second St.
San Francisco, CA 94107

Allow four weeks for delivery. All orders are sent by First Class Mail. California residents, please add 6½% sales tax.

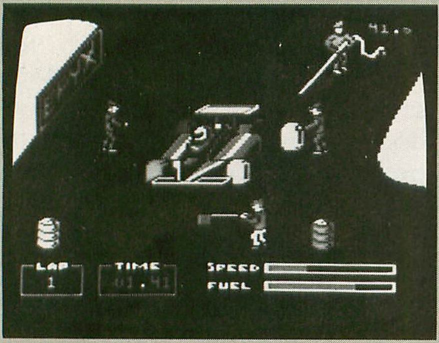
ANTIC presents these programs in diskette form for the convenience of the Atari community, in the belief that all of the programs offered are in the Public Domain and that no proprietary interests or rights to these programs are claimed by anyone. These diskettes are not copyprotected, nor does ANTIC claim rights to the programs themselves. The price of the diskettes is based on the cost of making them available.

The New Product Listings are compiled and written by Caitlin Morgan, Editorial Assistant. ANTIC welcomes submissions of products and information for this part of the magazine.

V-COS

(utility)
VERVAN Software
1953 W. 11th St.
Upland, CA 91786
(714) 946-5805
16K — cassette
$24.95

V-COS is a machine-language program that gives you more control over your computer's cassette operations, cassette baud rate, tape-leader length, screen background color, character-set color, and left and right screen margins. Simple keyboard commands verify cassette files and control the cassette's motor.



PITSTOP

(game)
Epyx, Inc.
1043 Kiel Court
Sunnyvale, CA 94089
(408) 745-0700
16K — cartridge
$39.95

When the urge to burn rubber becomes uncontrollable, a game of **Pitstop** might satisfy your craving. One to four players can experience the thrill of racing a Formula 1 car that comes complete with steering and speed controls. As the tires wear down and the gas supply dwindles, you must visit the pitstop. The tiny pit crew will get you back on the track in a flash. Decisions about when to stop and how fast to drive affect the outcome of this thrilling game of competition with the clock.

SUPER MAILER PLUS

(data-base)
Royal Software
2160 W. 11th Ave.
Eugene, OR 94702
(503) 683-5361
48K — diskette
$49.95

Created with both the novice and the sophisticated data-base user in mind, the foremost function of **Super Mailer Plus** is mailing list maintenance. While it is powerful enough for a small business, it is also flexible and user friendly.

ATARI BASIC — FASTER AND BETTER

(book)
IJG, Inc.
1953 W. 11th St.
Upland, CA 91786
(714) 946-5805
$29.95

Carl Evans, well-known to ANTIC readers as the author of "Tape Topics" and "Tangle Angles," has written an introduction to Atari BASIC. This book is the result of his attempts to get the most out of his computer with the least programming effort.

UNDERSTANDING ATARI GRAPHICS

(book)
Alfred Publishing Co., Inc.
P.O. Box 5964
15335 Morrison St.
Sherman Oaks, CA 91413
(213) 995-8811
$2.95

The company that created the famous series of Alfred Handy Guides has now added a number of computer-related titles to its list. This booklet offers practical, colorful lessons that will help you create graphics (including GTIA graphics), with either the 400 or the 800 computer.

Return the favor. When you call a manufacturer or supplier about a product you've seen advertised or otherwise mentioned in ANTIC, please tell them so. This will help us to continue to bring you the latest information about products that will make your Atari computer an even more valuable investment in the future. —ANTIC ED



RM 1000

(modem)
Macrotronics, Inc.
1125 N. Golden State Blvd., Ste. G
Turlock, CA 95380
(209) 667-2888
$239.00
Software, interface card and cables — $59.00

Ham radio operators and fans of the news/wire services will appreciate the **RM 1000**, a radio modem that sends and receives Morse Code and radioteletype signals. Multi-stage, commercial-quality demodulators are responsible for these feats, while the unit's unique, dual-bar tuning system brings in those signals loud and clear.

COMPLETE PERSONAL ACCOUNTANT

(application)
Programmer's Institute
P.O. Box 3470
Chapel Hill, NC 27514
(919) 967-0861
48K — diskette
$79.95

Formerly known as **The Color Accountant**, this renamed and redesigned program manages three major areas of personal finance: checking, budgeting and the scheduling of payments.

TAX CONSULTANT

(application)
Morbius Software Co.
Box 1702
Vacaville, CA 95696
(707) 422-9591
48K — diskette
$95.00

Designed for people with no tax experience, this complete program not only helps you prepare your annual tax forms, it also analyzes all entries and provides a printout of items that might be questioned by the IRS. **TAX CONSULTANT** is

capable of printing and preparing twenty-five federal forms and tax schedules, all of which are included in the program.

COMPUGARD

(home security system)
Tomorrow Tech
Box 308
5465 Wm. Flynn Hwy.
Gibsonia, PA 15044
(412) 433-8360
24K — diskette
$145.00

Now that your house is full of Christmas booty, are you afraid to leave it unattended? **Compugard**, a computerized home security system, can be your ever-watchful eye. Offering closed-circuit, perimeter protection, it operates via the CPU of your Atari.

MILLIONAIRE

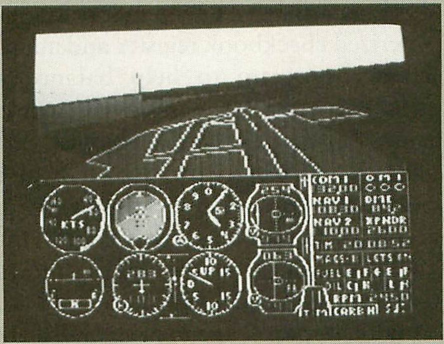
(game)
Blue Chip Software
19818 Ventura Blvd., #204
Woodland Hills, CA 91364
(213) 881-8288
48K — diskette
$59.95

Although you probably don't have $10,000 to invest in the stock market, you can live out your investment fantasies by playing **Millionaire**. If you *do* have loads of cash, you can practice buying and selling stock, maintaining a portfolio, and acting on market influences. Ninety-one weeks of stock activity are covered in this high-powered game of high finance.

THE TAX ADVANTAGE

(application)
Continental Software
11223 So. Hindry Ave.
Los Angeles, CA 90045
(213) 410-3977
48K — diskette
$69.95

Last year, did the ides of April find you furiously racing to complete your income tax form before midnight? If so, **The Tax Advantage** may spur you past such bouts with procrastination. The packet contains the standard 1040 form and all common, related tax schedules. Not only does it explain these forms in plain English, but it also displays the 1040 form on your terminal screen and performs complex calculations, such as income averaging. Don't forget, Uncle Sam is waiting!



FLIGHT SIMULATOR II

(game)
SubLogic Corp.
713 Edgebrook Dr.
Champaign, IL 61820
(800) 637-4983
48K — diskette
$49.95

No pilot's license is necessary for **Flight Simulator II**, a game that allows you to practice takeoffs and landings in a Piper 181 Cherokee Archer over 80 airports, in four areas of the U.S. Three-dimensional graphics, as well as changeable light and weather conditions, add to the game's realism. Dramamine is not included.

OVERCOMING COMPUTER FEAR

(book)
Sybex, Inc.
2344 Sixth St.
Berkeley, CA 94710
(800) 227-2346
$3.95

We assume that anyone reading this magazine has conquered computer anxiety, but perhaps you have a loved one who is plagued by the malady. He, or she, is not alone, as is pointed out in **Overcoming Computer Fear**. Written by the president of the Computer Clarity Group, Inc., the book is very accessible, and it provides both practical information and inspiration.

Our apologies to Dovestar Creative Concepts, the makers of **GRAPH-FIX**. In our November 1983 issue, we printed the wrong phone number in their New Products announcement. The correct number is (409) 727-5978.



THE WIRE AWAY

(storage device)
Networx
203 Harrison Place
Brooklyn, NY 11237
(212) 821-7555
$12.95

With the turn of four screws, **The Wire Away** can be mounted to the underside of your computer station, thus providing a safe place to stash loose wires. Four 18-gauge power cords (or 14 feet of excess wire) can be easily stored and concealed behind the unit's snap-on lid.

SHADOW WORLD

(game)
Synapse Software
5221 Central Ave., #200
Richmond, CA 94804
(415) 527-7751
32K — diskette & cassette
$34.95

Tricasmium is the most life-giving mineral in the galaxy — at least to the inhabitants of the planet Jantor. However, an army of invading spelunkers, the Rigillians, is planning to poison the precious stuff. You must lead the Confederation against the dastardly foe, despite a lack of space vehicles. On top of this small drawback, Jantor's unbearable climate spawns a most unfriendly, mutoid life-form, which may give you a problem or two. After playing this game, you may never complain about rush-hour traffic again.

COMPLETE PERSONAL ACCOUNTANT

Futurehouse, Inc.
310 W. Franklin St.
Chapel Hill, NC 27514
(919) 967-0861
$79.95, 48K – diskette

Reviewed by Tay Vaughan

The **Complete Personal Accountant** consists of ten programs for managing financial records, appointment schedules, and mailing lists. These ten programs are supplied on three disks; additional disks are required for the storage of data. A sizable booklet of instructions and examples provides detailed guidance for use and application of the CPA programs at home and in a small business.

Programs available are: chart of accounts, checkbook maintenance, checkbook search, summary budget analysis, detail budget analysis, net worth/income expense, payments calendar, appointments calendar, mailing list, and color graph.

Anyone who has started a small business or has had bookkeeping experience will recognize in these programs the essential ingredients of a common accounting system.

The Chart of Accounts is a list of the various categories in which money is spent or earned (categories include "salary," "rent," "medical bills," and even "penalties and fines"). A standard chart provided by CPA is divided into asset, liability, equity, income, and expense accounts. This chart can be easily customized by the user. Because all of CPA's financial processing programs refer to this list, the Chart of Accounts must be created and stored the first time the CPA program is used. Values are then posted to their proper category in the Chart of Accounts; 99 accounts can be listed, and each can include up to nine subaccounts.

Most people have had simple bookkeeping experience with their own checking accounts. CPA provides a computerized checkbook register and maintenance program to help balance a checkbook and keep permanent records of tax deductions, service charges, and deposits. Checks are automatically credited or debited to their appropriate accounts, and running totals are kept for all categories. A search option allows users to recall checks by number, date, amount, or payee, and to print or display them.

Budgets are also important for successful money management. CPA uses the Chart of Accounts to maintain a list of budgeted categories for the user, and also provides a general summary of budget status (with totals of budgeted amounts and amounts actually spent). A detailed analysis displays or prints budget items, and lists actual checks and cash expenditures by category.

If users faithfully update their data files, they can compute their net worth and generate a balance sheet at any time. Savings, stocks, bonds, investments, and other financial data that do not appear in checking accounts, are entered through a Net Worth/Income Expense program.

Two calendar programs allow users to keep track of dates when monthly bills are due, and to schedule appointments. The Mailing List program provided by CPA is useful for such things as Christmas lists or client directories, but is not as flexible as some other database-management programs. It is limited to names and addresses, although a 20-character field is provided for telephone numbers or other reference data. The Mail List prints only one name wide, so it can use friction-fed rolls of labels on the Atari 825, Centronics 737, or Epson MX-80 printers.

With the Color Graph program, users can produce a graph of any file that contains CPA financial data. Relationships between various accounts can also be shown graphically.

CPA is designed for a single-drive system, it has no option for multiple drive users. A great deal of disk swapping is therefore required. Error trapping is generally good, but not all decision branches in the menu-driven system allow for a return to the menu if the user changes his or her mind. Also, it is sometimes difficult to get an overview of a financial situation (such as the checkbook register) wtihout using the printer. Displays of checks and names appear only one at a time on the display screen. For a $20 fee, users can subscribe to a technical-support service from the publishers by telephone.

These are well-integrated programs that provide the backbone of a financial management system. Indeed, simply by following the step-by-step procedures outlined in the manual and faithfully entering the necessary data, many users will discover that their financial records are more organized and make more sense than ever before.

TAX COMMAND

Practical Programs, Inc.
P.O. Box 93104
Milwaukee, WI 53203
(414) 277-7378
$24.95, 16K – cassette
$24.95, 24K – disk
Add $2.00 for postage and handling

Reviewed by David Duberman

Do your knees start to quiver every year around tax time? Do April showers bring nightmares full of numbers and swirling sheets of paper? If you have an Atari computer, **Tax Command** may be able to help you cure your IRS headaches.

Tax Command is a two-part program written in BASIC. It's a very simple, but effective, computerized tax form that takes you through Form 1040 and Schedule A (Itemized Deductions) line-by-line and allows you to enter amounts where necessary. You can usually see the calculated results of your entries immediately.

The program is menu-oriented. You may select menu items by pressing a key that corresponds to a highlighted charac-

ter in the desired menu item. You need not press [RETURN] after selecting a menu item, and this is a real time-saver.

The main menu gives you a choice of the principal sections of Form 1040 — Total Income, Adjustments to Income, Deductions (Schedule A), and Tax Computation. The line number on Form 1040 is shown for every entry and calculated-result field.

Choose "Total Income," and you can enter figures for each line of Form 1040 that account for income. This category contains two pages of sub-menus, and you can return to the main menu at any time. "Adjustments to Income" works in a similar fashion.

If you choose "Deductions," you go to the deductions menu. Each item here corresponds to a category on Schedule A. Among these are Medical Expenses, Taxes, Interest Expense, Contributions, Casualty, and Other. There is a submenu for each of these, so every line of Schedule A is accounted for. Your total amount for deductions is constantly updated and displayed.

As you enter figures into Tax Command, you should also enter them by hand onto your tax forms. Tax Command has no provision for printing out a report, nor can you save the data. It's a program for one-time use — the tables and 1040 line numbers will probably be different next year, and this will necessitate a major revision of the program.

Once you've entered all your data, you activate Part Two. First, you're prompted to write down two numbers (your filing status and taxable income). Then you press a key to load the program. The loading process takes five minutes for cassette, but only a few seconds for disk.

Part Two asks you to enter the two numbers you previously stored in paper-and-pencil RAM, and then calculates your tax (or refund!) for you. Other menu items in Part Two allow you to change your filing status or add additional tax. I didn't have access to the 1983 tax tables at the time this review was written, so I was unable to determine the accuracy of the program's calculations.

If you press [G], you can enter your income for the previous four years. The program will then determine whether you can lessen your tax burden by income averaging (spreading out a sudden

continued on page 98

A CINDERELLA STORY

continued from page 92

simple. The problem is that the same color registers are used for plotting on the graphics screen and for text in the text window. The choice of certain plotting colors can ruin the text display. The table below and the demo in Listing 1 illustrate the problem.

The short demo first plots three bars that correspond to COLORs 1, 2 and 3. The program then changes the bar colors, using the SETCOLOR command, to illustrate how the text window is affected. Then it installs a DLI routine and again changes the bar colors to demonstrate that the text window is now independent of color changes in the graphics window. The demo will cycle between the two conditions until you press [BREAK]. The DLI gives you the freedom to choose any color you wish in Graphics Modes 3, 5 and 7. It works by resetting the color registers that control the text window to their normal values. This is done just before the electron beam scans the text window. You can gain the same independence in your own work by incorporating lines 150–200 into your programs.

IRQ'S OF INTEREST

In contrast to the NMI's, the IRQ's are not often used by the programmer. Two IRQ's of interest are the [BREAK] key and the keyboard. Any time a key is pressed, an interrupt is generated that causes the OS to process the signal coming from the keyboard. A knowledgeable programmer can redirect this interrupt so that certain keys are ignored. For example, you can disable the [BREAK] key so that it is ignored while your program is running:

```
A = PEEK(16):IF A> = 129
THEN A = A–128:POKE 16,
A:POKE 53774,A
```

Certain commands such as an OPEN to S: or E:, any GRAPHICS statement or a PRINT statement to the screen will re-enable the [BREAK] key. In those cases, you will have to repeat the above code as necessary. Note that pressing [RESET] will also reenable the [BREAK] key.

Many of the other IRQ's are used by the OS to perform I/O, as with the cassette recorder.

That's it for now. Until next issue, I'll leave you in the midst of the OS (I hope you're not *too* lost). In Part II, I'll cover the rest of the Operating System. Keep your flashlight handy.

Table 1

Basic Color Command	Basic Setcolor Number	Function Controlled in Graphics 3, 5, 7	Function Controlled in Graphics 0 (Text Window)
1	0	Graphics Points	—
2	1	Graphics Points	Character Luminance
3	2	Graphics Points	Background
0	4	Background	—

rise in income over several years).

Part Two's "Net Refund" menu applies if you've overpaid your taxes. Items include Tax, Other Taxes, Payments, Credits, and updated displays of 1040 lines 49, 56, and 68 (Balance, Total Tax, and Due IRS).

Tax Command is a straightforward and eminently practical program. You don't even need instructions — all prompts are clear and self-explanatory. If you dread the calculations that are needed to fill out your tax forms, it can save you a lot of trouble. It's not a fancy program, but it is reasonably priced. However, you still have to enter all data and make a few simple calculations by hand.

B/GRAPH

Inhome Software, Inc.
2485 Dunwin Dr., Unit 8
Mississauga, ONT, Canada
(416) 828-0775
$99.50, 48K — diskette

Reviewed by Jerry O'Neill

B/GRAPH is an important addition to the array of serious software available for Atari computers. It's a powerful program that allows you to enter data, perform many kinds of statistical manipulations, and make several types of informative charts and graphs automatically.

The graphs are displayed in a full-screen, high-resolution, four-color mode, and additional patterns can be used to distinguish different parts of the graph. B/GRAPH can save data as data files, and can save the graphs themselves as binary picture files. A "slide show" routine lets you display graphs in sequence. B/GRAPH also supports screen dumps to many popular printers.

B/GRAPH is a menu-driven program that "walks you through" the creation of graphs. It can create many different charts and graphs: point or line graphs, filled-in-area graphs, vertical bar graphs, pie charts and others. You select the graph you want, choose labels, and enter data in response to machine prompts. Once you've finished entering data, B/GRAPH draws your graph and labels it. If you use **VisiCalc** and have already stored data in DIF files, you can save significant time by entering these files directly. Utility programs load B/GRAPH charts into your own BASIC programs.

If you want to see the game data presented in different kinds of graphs, this is easy to do. For instance, you can switch directly from a bar chart with three-segment bars to a line graph. (Obviously, you can switch only between styles that use equivalent data.

I used B/GRAPH to create graphics from finished data. However, the program has a library of powerful statistical functions that can manipulate your data in ways that reveal important trends and correlations. A 33-page section of the manual covers these functions in detail. The manual, written by Ian Chadwick, author of *Mapping the Atari*, is complete and is designed for easy use, as is the program itself.

The statistical functions available include moving averages, geometric moving averages, exponential data smoothing, a t-distribution test, normal distribution probability, Poisson test, binomial distribution probability, standard deviation and many others, Suffice it to say that B/GRAPH knows far more about statistics than I do, and that it gives the user a comprehensive choice of functions.

Is B/GRAPH perfect, then? There were a few minor bugs in Version 1.0, but these have been cleared up in the current version 1.1. Unfortunately, 1.1 still contains some minor annoyances. When I was plotting a bar chart with only a few bars, for instance, I couldn't find any way to make the bars wider; as a result, I ended up with an empty-looking chart. I also managed to get "trapped" in the data-entry part of the program, and couldn't escape to the menu until I'd entered a few phony data values. (The program "knew" I hadn't entered enough points for a reasonable graph and wouldn't let go of me till I did.)

Finally, in the full-screen mode there are no command prompts on the screen when a graph is displayed; you'll want to create a reference sheet to help you remember key commands.

But these are all minor flaws. B/GRAPH's creators have given us a valuable program, and they have promised to provide as much continuing support as possible. (Incidentally, enhancement disks with extended features, such as support for additional printers and plotters, are in the offing.) All in all, B/GRAPH is a practical and extremely useful program. It's well worth the price, and both the program and the attitude behind it indicate that there *are* people out there who believe that the Atari is more than just a game machine.

THE HOME ACCOUNTANT

Continental Software
11223 South Hindry Ave.
Los Angeles, CA 90045
(213) 417-8031
$74.95, 48K — diskette

Reviewed by David Duberman

If your home finances are fairly complex, you should look into **The Home Accountant** for your Atari. But try to get a demo first — it's a powerful system, and is difficult to use, at least to begin with.

Fortunately, the 143-page manual leads you through the program in a step-by-step, logical manner. Unfortunately, the manual isn't quite thorough enough. I was left wondering about some of Home Accountant's features.

The manual doesn't mention it, but the autoboot, two-sided disk requires BASIC. Initializing a data disk takes three minutes. After this, you're prompted to insert side #1 of the program disk. The main menu boots, and off you go.

Not quite yet, though. The only menu item you can choose at this point lets you configure the program for your hardware

setup. Certain reports require 132 columns, and certain 80-column dot-matrix printers can print 132 columns in condensed mode. Home Accountant is quite flexible in this matter; the manual provides setup codes for a number of printers.

Next, you set up budgets. Up to 50 budget categories can be stored on one disk. First you establish one to five checkbook budgets. For each of these, you must enter the starting figure and the estimated or budgeted balance for each month of the year. After setting up each checkbook account, you must set up an associated cash account budget according to similar guidelines. This makes sense, since people often withdraw spending money from a checking account.

You then add budget categories of five types — assets, credit cards, liabilities, income, and expenses. Different but related budget categories can be started for complex expenditures such as mortgages — for instance, you can set up a liability budget for the mortgage principal, and an expense budget for the interest. If you establish a projected budget for each category, the program will calculate the difference between budgeted amounts and the actual figures.

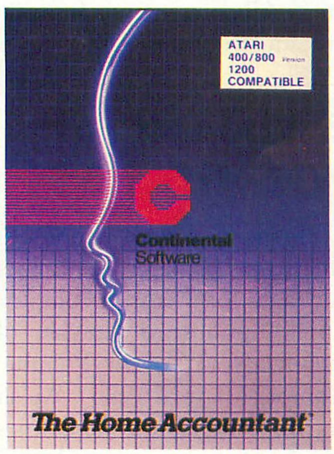
Budget categories of the income or expense type are handled differently. Instead of entering a beginning balance and estimating a monthly increase or decrease, you simply enter a monthly amount, which can vary depending on the category. The program then calculates a yearly total from the monthly figures.

You can change a category's name, type (unless it's a checking or credit card account) and amount entries, thus erasing the old category and creating a new one.

If you have a mortgage and a checking account with the same bank you may have arranged for an automatic monthly transfer of funds with the bank. Similarly, many accounts have automatic monthly deposits made to them. Home Accountant lets you use five automatic transactions per month per checkbook account.

After entering any automatic transactions, you enter all checkbook transactions. For each check or deposit, you enter the date, check number, payee, amount, a memo, and the appropriate category. You can also note (flag) whether an expense is tax-deductible, or whether a check has cleared the bank. The entry of credit card transactions is handled in a similar manner.

Next, you enter transactions for each cash account. You can number or classify each transaction with a six-character code. Tax-deductible cash payments may



be tagged as such.

A search/edit mode is associated with each type of transaction — checkbook, credit card, and cash. Other features allow you to reconcile your checkbook and bank statements, and to split transactions between budget categories, as in the mortgage principal and interest example mentioned above.

The Home Accountant can generate a variety of graphs and reports, including bar graphs that compare budgeted and actual amounts or trend-analysis graphs that feature linear-regression analysis. Graphs are plotted to the monitor screen, with an optional grid.

Three report categories are available: budget/actual, personal balance sheet, and income & expense summary. The first is important, but doesn't work well unless you have a printer with 132-column output. It provides a monthly summary of budgets and a list of transactions by category, as well as your projected and current net worth and total income. Other reports summarize actual monthly balances for different categories, note differences between budgeted and actual monthly expenditures, compare totals by month, and print income and expense summaries. Reports can be generated on your printer, and you can print out checks on special forms that are available by mail-order.

The Home Accountant is a bit bewildering at first, but with time and experience most of its functions will become understandable. If you have the need, the patience and the proper hardware (two drives are a considerable advantage with this system), this program can pay for itself within months. By eliminating the need for a complex bookkeeping system and by automating data entry, it should save a typical middle-class householder many hours of tedium each month.

PERSONAL ACCOUNTANT

Softsync, Inc.
14 E. 34th St.
New York, NY 10016
(212) 685-2080
$34.95, 48K — disk

Reviewed by David Duberman

Softsync's **Personal Accountant** home finance program is touted as an easier-to-use alternative to Continental's **Home Accountant**. It is easier to use — albeit a good deal simpler in content.

Instead of budget categories, in Personal Accountant you set up accounts. It comes with 12 accounts, including checking, charges, rent, leisure and food. You can delete or rename any of these, or add your own. There are four basic types of accounts: deposit, asset, liabil-

continued on next page

ity, and expense.

You can store as many as 144 categories on a disk, because each entry doesn't contain much data. In fact, all you enter for each transaction is the date, a code (of up to eight characters) and the amount involved.

Personal Accountant helps ensure that your books are balanced by employing a system called double-entry. For each entry, you must specify two accounts. For instance, if you start with a paycheck for $1000, you enter income as the first account and checking as the second account. If you enter an expense, the amount is posted to the proper expense account as a positive number, and to the paying account as a negative number. One type of report, the trial-balance sheet, lets you keep track of whether amounts are being properly entered by comparing balances.

Other functions of the menu-driven program allow you to read a file or part of a file, and to change or delete an entry. If you can't remember which entry you want to change, you can review a list. You can also list all files on the disk, although this process can be somewhat slow. Another problem, at least for beginners, is that you have to RUN the program from BASIC, and the instructions don't explain the procedure very well. In addition, the [BREAK] key is not disabled — another difficulty for beginners. Also, the rather sketchy documentation fails to mention that a press of [RETURN] in response to many prompts configures output for a minimum system (i.e., no printer). This can save the user considerable time. The program's error trapping is good, but not great. If an amount contains only one decimal place, that's how it is printed — a marked deficiency, especially with tabular output.

Although a disk can contain thousands of entries, the program only accounts for the eventuality of a full disk by allowing you to forward account balances to a new disk. Individual entries cannot be forwarded.

One nice feature of this program, which is not mentioned in the documentation, is that all its reports are no wider than 40 columns. As a result, it can be used with any printer. In addition to the trial-balance report mentioned earlier, you can generate an expense-accounts report that lists all expenses by account and provides a total. You can also print a listing of assets and liabilities (with a balance of Assets over Liabilities) and income-and-expense recap (with an Income over Expenses balance).

Two other programs, which also must be RUN from BASIC, are present on the program disk. Program A calculates loan payments, amortization tables, future values, and future values with annual interest. All data entry is from the keyboard, and output is to the screen only.

A FINANCIAL WIZARD 1.5S

A Financial Wizard is one of the oldest and best-know home finance packages for the Atari computers. ANTIC published a review of A Financial Wizard in Volume 2, Number 1 (April 1983). We summarize that review here for the benefit of our newer readers.

A Financial Wizard makes good use of the Atari's sound and graphics capabilities in a complete package for managing your home finances. You can enter budget data, categorize checks, produce graphics on-screen, print checks, and use many other functions. The new version, 1.5D, adds two new report-generating programs.

A Financial Wizard is available on disk for $59.95 plus $3 postage and handling from:

On Line Computer Centers
of OKC
10944 North May
Oklahoma City, OK 73120

The final program, called N, is a simple name, address, and phone number data base. You can search for entries by stipulating the first characters of any of the five lines in the entry, and can print name-and-address labels. Despite advertised claims, this program isn't really integrated with the rest of Personal Accountant.

This program may be useful to those who are just starting out in home finances and who may not need Home Accountant's exhaustive thoroughness. The documentation is adequate, but prompts are sometimes less than helpful. If you can see a demonstration before you buy, by all means do so. If that's impossible, consider your needs. If financial problems are starting to pile up, Personal Accountant may be the answer for you.

FINANCIAL ASSET MANAGEMENT SYSTEM

Atari Program Exchange (APX)
P.O. Box 3705
Santa Clara, CA 95055
(408) 727-5603
(800) 672-1850 (CA);
(800) 538-1862 (outside CA)
$29.95, 40K — diskette

Reviewed by Karl Wiegers

Financial Asset Management System is a comprehensive program designed to monitor and evaluate your portfolio of financial assets and investments. It features easy file and record-handling capabilities and can print several different kinds of financial reports. A clearly written 12-page manual (with appendices that illustrate sample reports) is included with the program.

The user of Financial Asset Management System first creates a file to contain asset data. This file can hold as many as 99 individual assets, with a total value of up to $10 million. Additional files can be created as they are needed.

You assign each of your assets a type

number from 1 to 99, depending on whether its quantity is measured in shares (1–70) or dollars (71–99). This gives you the degree of flexibility needed to handle such diverse assets as stocks, bonds and mutual funds (measured in shares), or bank accounts, property and collectibles (measured in dollars).

To create an asset record, you enter the asset's name, a six-character code, its type number, and data that include its quantity (in shares or dollars), initial cost, current price, and expected annual payout per share. A menu of maintenance operations makes it easy to add, delete or update assets in this file. Functions such as updating prices, quantities, costs and payout rates, or entering dividends (paid or reinvested), can also be performed. A handy "zero all dividends year-to-date" feature is helpful for income tax purposes.

Financial Asset Management System can print four kinds of reports on Atari 825 or Epson printers. The most useful of these lists assets alphabetically within type, providing the initial data you entered along with current total value, percent of total assets, profit in dollars, percent profit, annual payout and yield. Subtotals by type and a grand total are also printed.

I update my asset data and print this report on a weekly basis to track my investments and see what changes I should take into consideration as economic conditions vary. Other reports include: lists of assets sorted by magnitude of total value, profit and other parameters; a listing of year-to-date dividends paid; and a listing of the entire contents of the file (without the calculated data that appear in the first kind of report). A convenient customized-data input form can also be printed to facilitate the updating of your own data file.

This program uses virtually no sound, graphics or color, but it completely lives up to the implicit promise made by its title. It is easy to use, appears to be bug-free, and produces attractive, effective printed reports. The only difficulty I've encountered with the program involves the tracking of municipal bonds, whose prices change as interest rates fluctuate. To handle them, I've had to recalculate current bond prices and then enter the new values into the computer. Otherwise, however, I've found Financial Asset Management System to be enormously helpful in financial and tax planning. You still have to do your own investment analysis, but this program greatly facilitates the tedious task of tracking your assets.

THE MONEY PROCESSOR

Luck Software
1160 Niblick Rd.
Paso Robles, CA 93446
(805) 238-2585
$59.00, 48K – disk

Reviewed by Dave Mentley

The Money Processor is one of those rare programs that will actually justify the expense of your computer hobby. A combination spreadsheet and database program, the Money Processor is an efficient and easy-to-use personal financial-record keeper. Ease of use is guaranteed by its screen editor, one of the most well-engineered user-interface devices I have seen in any software package. And its machine-language code is written so tightly that you won't even know the program is running. In addition, an Introduction, Operation Guide, Owner's Manual and a unique keyboard overlay are included to help you make full use of the package.

Essentially, the Money Processor is a filing system and worksheet for all transactions that involve spending or earning money. On one statement, you can consolidate data about all of your check deposits, checks drawn, credit-card charges and payments, savings-account transactions, cash spent, and employee expenses. You can also build a list of tax return and household budget items. The program uses the actual names of your accounts, rather than their numbers — the tedious method used by most other personal-finance packages.

To begin, you must enter the name of your accounts; you can enter or delete an account at any time. The program allows for up to 150 accounts in 7 categories: Credit Cards, Checking Accounts, Savings Accounts, Cash, Employee Expense, Tax Return Items and Budget Items. Once you have entered your account names, you need to enter data on all relevant transactions from your records on an account worksheet. Any entries you make are stored on your data disk; if you move from one screen or a worksheet to another, your data is saved automatically.

All of the program's worksheets are customized. "Credit Cards," for example, is set up to handle charges, payments, miscellaneous charges (fees) and miscellaneous credits (returns or refunds). When you've entered all appropriate transactions for your statement period, you can generate a statement. This makes each transaction as "verified" and includes it in a dated statement that can be printed and filed. These statements are also maintained on disk.

Balances from each statement (Date) are fed into a balance for each account (Visa, MC), which in turn is fed into a balance for each type of account (Credit Card); this lets you check your financial condition at any time. The closing balance from the last statement is transferred as the "Last Statement Balance" to your next statement to give you current balance.

The Money Processor offers a number of features that make it easy to use and virtually foolproof. The editor with which you enter data from the keyboard, in particular, is exceptional. It locks you out of any area in which you're not allowed to type, and permits you to enter only numbers (no letters) into the Data and Amount columns. The screen is 54 columns wide, and scrolls as you move the cursor. This gives you room for extensive comments on each transaction.

continued on next page

The controls also allow you to move across the worksheet very quickly.

Perhaps the most fascinating feature of this program is that its total balance is updated each time you enter a digit in the Amount column. In addition, a window at the bottom of the screen always tells you how much space is left in memory and disk. Since you start with more than 350 lines for accounts and worksheets, you have more than enough space, even if you are a tremendously prolific spender. In fact, the program's data-compression techniques allow you to pack more than a year's worth of transactions onto a single disk.

To eliminate the need to memorize the program's entry and edit functions, Luck has designed a keyboard overlay that fits your 400, 800 or 1200 console and reminds you of the key functions. This overlay, in combination with the Introduction, Operation Guide and Owner's Manual, provide thorough documentation for the program.

The Money Processor will save you countless hours of tedious work at tax time: You can boot it up and do two months' worth of record keeping in half an hour. It also helps you track your expenditures and check your bank's arithmetic. It sets a standard of excellence for other manufacturers of commercial software to emulate, and I highly recommend it.

PHARAOH'S PYRAMID

Master Control Software, Inc.
P.O. Box 26714
Salt Lake City, UT 84126
(800) 624-5596
$34.95, 16K — cassette
$34.95, 48K — disk

Reviewed by Lawrence Dziegielewski

The setting is ancient Egypt, the civilization that blossomed on the banks of the River Nile. The days are hot, the nights are cool, and the Great Pharaoh Rameses has summoned you to build a pyramid for his eternal resting place. "Do this," he says, "and the riches of Egypt are yours! But beware the demons who will pursue your every step. They will not cease until they have stopped you. Good luck, Little Achmed!"

In **Pharaoh's Pyramid**, you are Little Achmed, the builder of pyramids. Six difficulty levels move you through the ranks of Egyptian society, from slave to Pharaoh. If you complete the tasks of level 6, and finish the pyramid, Rameses can be laid to rest, and you can claim your reward.

To complete each level, you must maneuver down a series of steps to the Nile, gather up the bricks that lie there,



and return them to the top. In the process, you must avoid many enemies, including Moses's Snake, four separate plagues, and the cunning Osiris, who appears suddenly during each round. Touch any of the demons, and you lose a life.

Pharaoh's Pyramid is very challenging, even at the easiest level. The demons pursue you with dogged determination, and only speed and agility allow you to avoid them. This game definitely puts your reflexes to the test!

You can start the game at any of its six levels. A two-player option can be selected by pressing [T].

I found the game's graphics to be excellent, and its screen is both colorful and detailed. Superb use of sound and music also enhance its appeal. In addition, Pharaoh's Pyramid has staying power, that rare ability to challenge you each time you play.

I was hard pressed to find a flaw in this game, but I did note one small quirk. Moving Achmed around on the steps requires diagonal joystick input. As a result, if you use a worn out set of joysticks, you must make very precise movements. I found this to be frustrating at times. However, a good quality joystick greatly enhances the playability of this game. When I borrowed a friend's WICO joystick, my scores increased noticeably.

Pharaoh's Pyramid is a very entertaining and worthwhile game. If you're a serious gamer, and aren't afraid of a good challenge, this game should be part of your library.



POOYAN

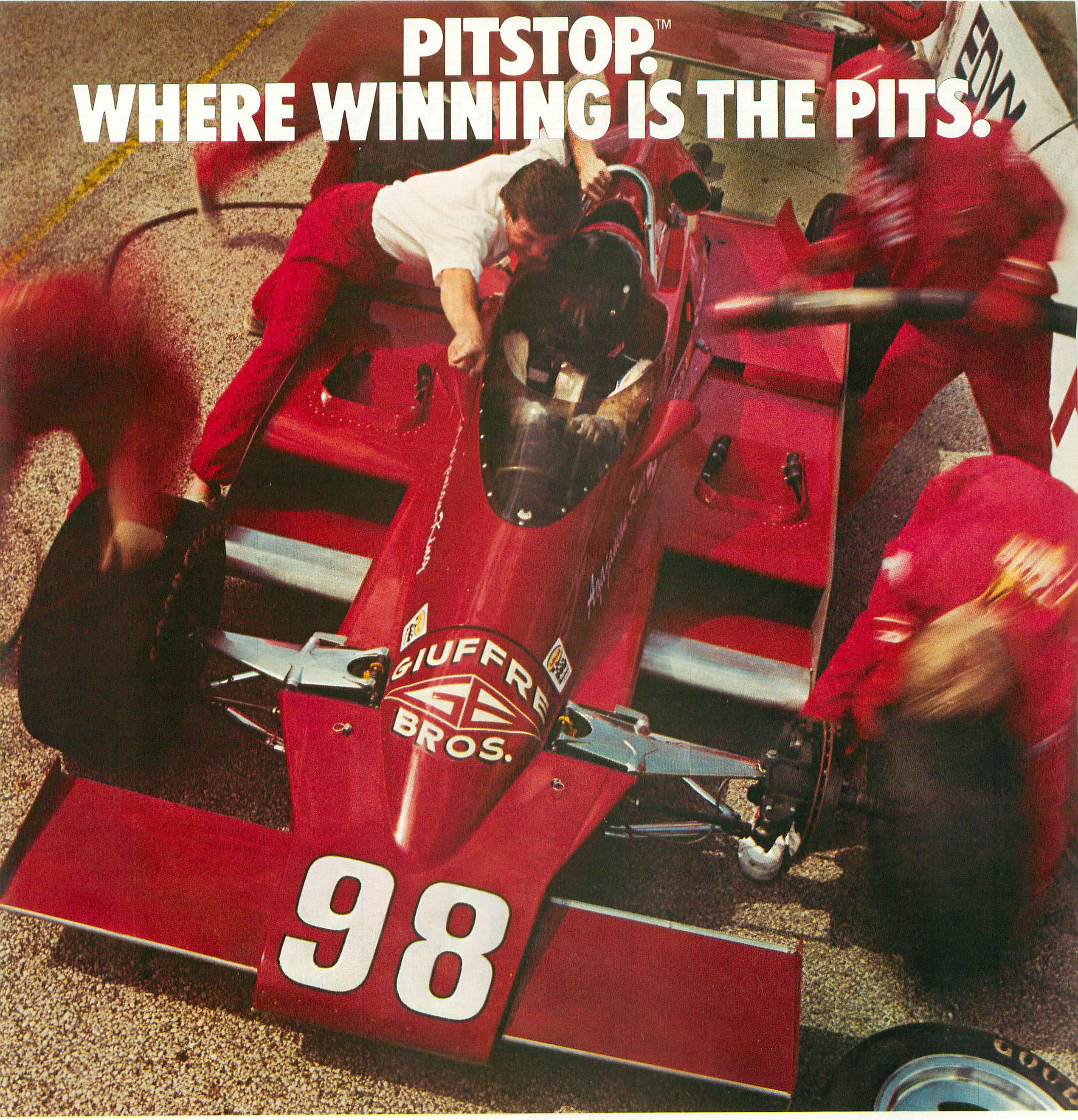
Datasoft Inc.
9421 Winnetka Ave.
Chatsworth, CA 91311
(213) 701-5161
$29.95, 32K — disk & cassette

Reviewed by Mark Cotone

As one of the higher members of the food chain, I've never really appreciated the constant battle for survival going on around me. But I got an education in survival from **Pooyan**, a Konami game recently released by Datasoft.

You probably remember Konami for past arcade hits like Frogger, Scramble and Time Pilot. Pooyan, a relative unknown, is a revised version of the "Three Little Pigs" story. You play the role of a courageous parent pig who fights to protect home and family from the clutches of the big, bad wolves. Now, who's afraid?

continued on page 104





You'll never make Grand Prix champion just driving in circles.

You've got to stop sometime. The question is when. Right now you're in the lead. But the faster you go, the more gas you consume. And the quicker your tires wear down.

If you do pull into the pits, though, you lose precious seconds. So it's up to you to make sure the pit crew is quick with those tires. And careful with that gas. Otherwise, poof! you're out of the race.

See your retailer for available computer formats.

So what'll it be, Mario? Think your tires will hold up for another lap? Or should you play it safe and go get some new ones?

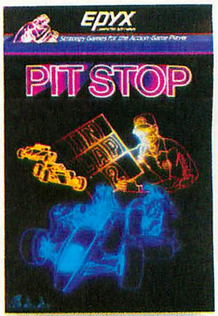
Think it over. Because Pitstop™ is the one and only road race game where winning is more than just driving. It's the pits.

Goggles not included.

One or two players; 6 racecourses, joystick control.

EPYX COMPUTER SOFTWARE

STRATEGY GAMES FOR THE ACTION-GAME PLAYER.



The game begins at the pigs' home, a sturdy dwelling built into a rugged mountainside. You patrol the face of your cliffside abode in a joystick-controlled, armored basket. Suddenly, from out of the nearby treetops, a pack of hungry wolves mounts an attack. Clinging to helium balloons, the carnivores attempt to float down onto your front porch. They have dinner planned, and your babies are the main course!

Taking careful aim, you fire arrows at the wolves' balloons. If you succeed in popping the balloons, you send the wolves, tumbling end over end, to a ghastly death. But these wolves aren't sitting ducks; they're armed with deadly acorns. One touch from these nuts and your basket will topple, sending you on your own deadly freefall. Your secret weapon, a slab of beefsteak, should be saved and hurled into the pack of floating attackers when the moment seems right. Each wolf will then let go of its balloon to grab at the meat.

When you master this level of play, the skill level changes automatically. Several new twists and bonus scenes are added to spice up the action.

What separates Pooyan from all the other arcade shoot-outs? The main distinction is that, although many games can match Pooyan's fine, detailed, graphics, few are as well animated. The wolves don't simply move across the screen — they strut with menacing smiles.

When a wolf leaps from a treetop, he appears to be frightened at his vulnerability, and frantically eyes the onrushing earth. The quality of animation is superb. Good graphics, sophisticated player movement, and challenging gameplay — all the ingredients of a winner — are evident here. Pick up Pooyan for yourself, and fight for your life. You'll be glad you did.

ACTION!

Optimized Systems Software, Inc.
10379 Lansdale Avenue
Cupertino, CA 95014
(408) 446-3099
$99.00, 16K—cartridge

Reviewed by Jerry White

ACTION! is a complete software development system, written by Clinton Parker of ACTION! Computer Services and distributed by OSS. ACTION! is a structured language on a bank-select 16K cartridge. The cartridge also contains the monitor, compiler, library routines, and the most sophisticated editor I've seen to date. The bank-select feature built into the cartridge ensures that ACTION! will use only 8K of your precious RAM.

For those who have found BASIC to be too slow or assembler too difficult, ACTION! is the logical alternative. ACTION! programs can increase speed to from 50 to 200 times that of BASIC. You will find that ACTION! performs nearly as well as assembler in terms of speed and compactness of object code.

BASIC programmers will be glad to know that most Atari BASIC commands are included in ACTION!'s procedure libraries. ACTION!'s vocabulary also includes AND, ARRAY, BYTE, CARD, CHAR, DEFINE, DO, ELSE, ELSEIF, EXIT, FI, FOR, FUNC, GET, IF, INCLUDE, INT, LSH, MOD, MODULE, OD, OR, POINTER, PROC, RETURN, RSH, SET, STEP, THEN, TO, TYPE, UNTIL, WHILE, XOR, and more.

The Library is a collection of subroutines that can be used in a program, but do not have to be defined within that program. These subroutines are not actually part of the ACTION! language. They are provided for your convenience.

When you turn on your computer with the ACTION! cartridge installed, you start in the Editor. The Editor treats

your text as though it were on a long sheet of paper. The display is treated as a window through which you can see only part of the text (one screen). The Editor has commands that allow you to move this window anywhere on the "paper." Inside the window you can make changes to the text.

You can search for a piece of text, and you may optionally substitute another piece of text for the one you find. It is possible to leave an invisible marker (tag) at the current cursor position, and then go back to that tag in the future without having to search for it.

Last but not least, you can have two of these windows (look at different "sheets of paper") on the screen at once. This feature makes it possible to move pieces of text from one "sheet of paper" to another. You can also enter and save your text using any supported I/O device.

Numeric variable support includes BYTE (0–255), CARD (0–65535), and INTEGER (–32768 to 32767). Floating point is not directly supported. ACTION! even supports immediate mode commands, user-generated procedure libraries, and the ability to intermix 6502 machine language instructions.

I wrote a couple of short programs to see how fast each could simply count to 10,000. BASIC completed the task in 1164 jiffies while ACTION! required just over 11 jiffies (sixtieths of a second). While other benchmark tests showed slightly less than this ratio, some others have shown ACTION! to generate code over 200 times faster than BASIC.

FUN WITH ART

Epyx (Automated Simulations, Inc.)
1043 Kiel Court
Sunnnyvale, CA 94086
(408) 745-0700
$39.95, 16K – cartridge

Reviewed by David Plotkin

One problem I've always had with graphics drawing programs is that I never remember the multitude of command keystrokes needed to achieve the results I want. Enter **Fun With Art** (FWA) from Epyx. Not only is this the first cartridge-based graphics program (most require a disk drive), but its commands can be accessed by pointing to icons (symbols) on a command screen. Add some powerful features not found elsewhere, and you have an excellent package for the advanced and the novice artist alike.

FWA allows you to create pictures in ANTIC Mode E, the high-resolution, four-color graphics mode that is sometimes referred to as Graphics 7½. All drawing is done with your joystick. Many common commands are supported, including LINE, POINT, COPY BLOCK, and MOVE BLOCK. There are two ways to change drawing commands – either enter the appropriate letter from the keyboard, or, if you're like me and can't remember the proper letter, press [START] to view the command screen. Then use the joystick and fire button to select the icon that represents the desired command. The name of the command under the cursor appears in the text window, along with the command currently in effect and various instructions.

This program offers a number of unique capabilities, in addition to the icon menu. For example, you can assign a priority number (from one to four) to each of the four colors, including background, so you won't accidentally draw over a color in a tight spot. The background color must be given highest priority if you want to erase a line or a small portion of your drawing.

FWA also allows you to load and save partial screens (color forms), but these operations require the use of a disk drive. One of the features I miss, though, is "pattern fill," the ability to fill an odd shape with color. FWA's fill functions work in any of four directions but cannot easily fill an odd-shaped area. Another minor complaint: FWA's error messages are very cryptic, and appear in hexadecimal form!

The program's most impressive feature is that it allows you to change the four basic colors at every other scan line on the screen by using Atari's Display List Interrupt function. This makes for a lot of colors! The display list is saved with the picture to disk, so its array of colors becomes an integral part of the picture.

Finally, Epyx provides good documentation for FWA, including an explanation of how to load and display your pictures with a BASIC program. Full instructions on memory management and a set of relocatable machine-language routines are also included. All in all, FWA is a lot of fun to use. It is an excellent stand-alone package.

EXCALIBUR

Atari Program Exchange (APX)
P.O. Box 3705
Santa Clara, CA 95055
(800) 538-1862 (national toll-free)
(800) 672-1850 (CA)
$29.95, 48K – disk

Reviewed by David Duberman

"Oh no!" I thought when I received my review copy of **Excalibur**. And there was good reason for my panicky response. The disk was accompanied by more documentation than I had ever seen with a game program – some 80 pages, including a 69-page novel about King Arthur!

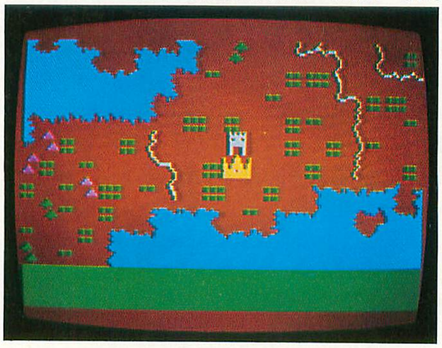
But Excalibur easily ranks as the finest programming achievement to date by Chris Crawford (the designer of such state-of-the-art war games as Eastern Front 1941 and Legionnaire). In this fantasy simulation, set in Medieval England, you assume the role of Arthur, King of Camelot and rightful heir to the throne of all Britain. You've recently pulled Excalibur, the magic sword, from the stone. To win, you must conquer or otherwise win the loyalties of the other 15 kings in Britain.

As with Crawford's other recent ef-

continued on next page

forts, real-time battle is a strong factor in Excalibur. In this game, though, it's only one of many factors. Your world also includes economics, diplomacy, friendship, loyalty and a little magic. When the need arises, Merlin, you trusted counselor, can visit plague or pestilence upon an enemy, or turn the heart of an unfriendly king in your favor. His best trick is *See*, which lets you surreptitiously visit any king to discover his vital defense secrets.

Execution makes this program special. An exceedingly complex game (perhaps the most complex yet written for an eight-bit computer!), Excalibur's performance often approaches the level of Artificial Intelligence (AI). Despite its complexity, however, practically all of its functions are controlled with one



joystick.

After the game boots (with some graphically stunning titles), you find yourself, as Arthur (represented on the screen by an icon of a gold crown), in a vertical corridor in your castle. During the course of the game, you visit different rooms that pertain to various functions of your role.

In the Room of the Round Table, you monitor the behavior of your knights, reward them or honor them as you deem fit, and banish them if necessary. Here, also, you select the knights who will accompany you in battle.

In the Throne Room, which is furnished with a Magic Map of Britain, you follow news of the kingdom, chart the rise and fall of your own prestige, and

continued on page 108

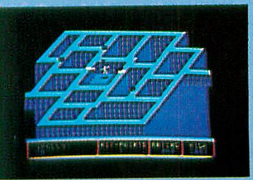
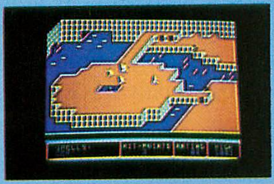
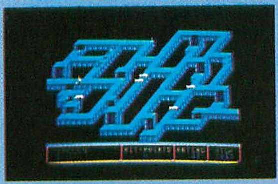
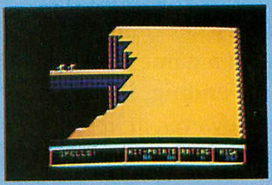
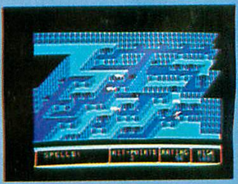
ZOMBIES©



*Atari is a trademark of Atari, Inc.

© Copyright 1983 BRAM Inc.

Scrolling 3D graphics, on-line instructions, one or two player cooperative, seven different dungeons, 74 different screens, high score save to disk, full sound and color, zombies, poisonous snakes, giant spiders, evil orbs, scrolls, talismans, magic spells, lost crowns and spectacular underground scenery.

A fast action arcade fantasy for Atari's.*

By Mike Edwards from BRAM Inc.

Disk and Tape
$34.95

BRAM Inc.
18779 Kenlake Place N.E.
Seattle, Washington 98155
(206) 486-8428

have the opportunity to either declare war on or pay tribute to opposing kings.

The Treasury Room offers you a vision of Camelot's wealth and of the tithes that you receive from your vassals. Here you preside over the taxation of your subjects and determine the size of your army.

Finally, in Merlin's Room, you exercise the magic powers mentioned above.

You exit the castle from the lower edge of the Throne Room. The disk spins, and you find yourself on a scrolling map of Britain. Press a button, and Merlin appears in the form of a raven. Guide the raven to any castle, and you will learn its name and the number of its men-at-arms. If you're attacking, you can pillage your enemy's crops. You can also enter a vassal's fortress and help him to battle an intruder.

If you attack or are attacked, the two opposing armies are placed on a battlefield. Those of you who are familiar with Crawford's war games won't need any further instruction here — control of your men is straightforward and is well suited to the realistic action. You may attack or retreat at any time during a battle. When the fight is finished, wins and losses are tallied. One final note: A game can be stored at any point and resumed at a later time.

This project was conceived and initiated by Crawford, but he gives co-author credit to Larry Summers and Valerie Atkinson, who started out as his assistants. Their contributions have unmistakably broadened the game's scope; all three are to be heartily congratulated for an awesome achievement.

Oh yes, I almost forgot the novel! It follows Arthur from his humble beginnings through an entire game, and includes many concrete strategems that you may want to use in a game of your own. I won't reveal the ending, but it is a happy one. You are, however, at liberty to create a new scenario in every game.

Chris Crawford describes this game as "not candy" — an accurate assessment. It's impossible to play Excalibur with the mindset you might use for a game of Pac-Man, for example. You have to read the novel (and re-reading it several times wouldn't hurt), you have to keep notes, and you have to think (a lot). It takes time to learn to play Excalibur well. But it's an investment that can pay off in one of the richest gaming experiences ever.

MAGIC STORYBOOK: THE THREE LITTLE PIGS

Amulet Enterprises
P.O. Box 25612
Garfield Heights, OH 44125
(216) 475-7766
$19.95, 16K — cassette

Reviewed by John and Mary Harrison

"I'll huff, and I'll puff and I'll blow your house down!" shout Keith and Wasco. Keith is our three-year-old son; Wasco is the wolf in **Magic Storybook: The Three Little Pigs**. At last, there is some entertaining software available for the young child! This program takes the form of an animated-video story book with a cassette-based sound track.

There is much to like about Magic Storybook. It is beautifully drawn and animated. The pigs' ears wiggle, stars twinkle, men race across the screen, and the houses shake as Wasco blows them down. The music is excellent. It helps pass the time during the four-minute load and announces the arrival of Wasco in each scene. The narrator's voice is clear, soothing and pleasant. The voices of the three pigs and Wasco are distinct, and each fits its character.

In addition, for parents who are concerned about the violence that is inherent in all fairy tales, the script minimizes the wickedness of the wolf. At the end, when Wasco lands in the boiling pot, he is not "killed." He is merely "gone." Most importantly, we found that when the program was viewed by neighborhood children aged two to four, it was a resounding success. They played it over and over.

Unfortunately, Amulet chose to market this product as "entertaining educational software for children of all ages." We object to Amulet's use of the word "educational."

There are many opinions about what constitutes educational software. Yet, at a minimum, a piece of software cannot be considered educational unless it (1) has a target audience in mind, (2) clearly states one or more educational objectives, and (3) includes a guide that explains how the program can be used to attain these objectives.

It is important to identify a target audience so that parents know they're not buying something either too simple or too complex for their child. Magic Storybook appears to be most appropriate for children aged two to five. Our three-year-old son loves it, but it doesn't provide enough for our six-year-old to do.

In fact, Magic Storybook has little educational value. It merely relates the classic story of The Three Little Pigs, much as you might tell it to your children. It may keep their attention and they may find it enjoyable, but what have they learned? There are no skills taught, no new concepts presented. Even if a child is completely passive, the program automatically moves Wasco to the proper place after a one-minute pause. Perhaps the program promotes listening skills, but then, so does a record book. Finally, there is nothing in the documentation that indicates an intended educational use of the product. The package contains a copy of the script, loading instructions, and a picture to color, but no guidance for the parent is included.

Magic Storybook does not fit into any existing category of home computer software. It competes with the record books that we received as children. The educational value of the product could easily have been increased by including a game based on the story. However, this *is* one of the few products on the market that is aimed at very young children, and at $19.95 it is competitively priced. While it can be used to introduce very young children to the computer, enter-

tainment, not education, is Magic Storybook's major contribution.

SIGNALMAN MARK XII MODEM

Anchor Automation, Inc.
6913 Valjean Ave.
Van Nuys, CA 91406
(213) 997-6493
$399.00

Reviewed by David Duberman

Anchor Automation has become a familiar name to the microcomputing community in the past year, largely as a result of its highly successful Signalman series of 300-baud modems. Inexpensive and easy to use, Signalman modems have helped introduce thousands to the joys of telecommunications.

With the introduction of the **Signalman Mark XII**, Anchor brings its reasonable prices and telecommunications expertise to the 1200-baud market.

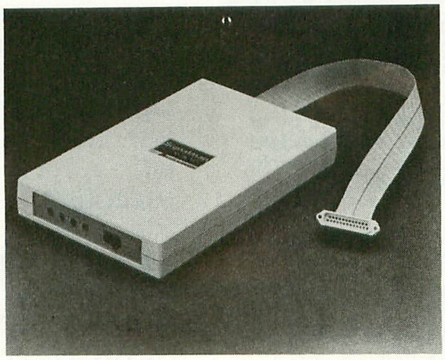
Until now, the only 1200-baud modem commonly available for most microcomputers has been the Hayes Smartmodem 1200. Hayes has, by default, set a standard for modem operations that, fortunately, has been followed very closely by the engineers at Anchor.

Installation of the Mark XII is quite simple. A modular cord that connects the unit to a wall outlet is supplied, as is a 110-volt AC power adapter; there is no provision for battery power. A 25-pin RS-232C male interface cable is permanently attached to the unit. An optional gender-changer gives you a female connector as well. Unfortunately, neither of these connects directly to the nine-pin RS-232C connector on the Atari 850 interface module, and Anchor doesn't make such a cable.

However, if you have a cable that can connect a Hayes modem and the 850, you're in luck. Such a cable works with the Mark XII without modification. Otherwise, you must make a cable or have one made for you. Nine pins need to be connected for full functioning (including auto-answer), or six for minimum functioning. "Appendix A" of the Mark XII manual lists the applicable pins and their functions.

In *situ*, the Mark XII works much like the Hayes. However, it doesn't feature as many LED's — just one each for High Speed, Carrier Detect, Send/Receive Data, and Modem Ready. Also, it does not include a built-in speaker for monitoring calls. Instead, the monitoring of current status is performed by the firmware in the modem. If, for example, you attempt a call with no success, a message flashes on the screen after a 10-second-pause.

The Mark XII operates in either of two states. In normal operation, using a terminal program such as Datasoft's Tele-Talk, you're in the command mode as soon has the program has booted. At



this point, you can command the modem to dial a number (ATDTnnnnnnn), or to answer the phone automatically when it rings. For instance, if you type ATSO = 1, the modem will answer the phone after one ring. Then it will transmit a carrier signal in an attempt to establish communication with the modem at the other end of the line. The Mark XII's command language and syntax are identical to those used by Hayes.

Once contact has been made with another computer, you are placed in the on-line modem and may proceed as usual. To return to the command state, press [+] three times quickly. "OK" will appear on the screen, and the Mark XII will be ready to accept commands. To return on-line, use the command [O].

The Mark XII operates in either full-duplex or half-duplex modes. Full duplex provides for the simultaneous transmission of data in either direction (like a voice telephone line), while half duplex requires that one party wait to transmit until the other is finished. Also, the modem operates at 110, 300, or 1200 baud, terms that apply to the rates at which data is transmitted. The manual briefly describes the implications of these different modes of operation.

It also describes the many different command codes available for programming the Mark XII. Among these are A/ (last command repeat), I1 (ROM test), and H1 (off hook). There seem to be sufficient commands to cover any conceivable situation involving telecommunications. A list of commands, along with brief explanations, appears on a pocket-sized reference card that comes with the modem. The card also lists results codes (modem responses to your commands) and set registers. These let you control certain of the modem's variable parameters, such as the number of rings to auto answer, the escape-code character (used to return to the command state from the on-line state), and which characters indicate a line feed or a back space.

At a price that is some $200 less than that of the Hayes, and featuring almost identical functions, the Signalman Mark XII modem seems to be a logical choice for Atari owners who require, or expect to have use for, a 1200-baud device. It should be noted that pin 20, which monitors the status of the Data Terminal Ready (DTR) line, is not used by the Mark XII, although it is used by the Hayes. This is of minor significance, however, and will not affect operations in most cases. Also, the Mark XII manual, while complete, is not as user-friendly as the Hayes manual. If you're a newcomer to telecommunications, you will probably be left with many questions after reading the Mark XII manual. Hopefully, the folks at Anchor will bring out an improved and revised version of the manual sometime soon.

ADVERTISERS

This is provided as a convenience and as a courtesy to advertisers. ANTIC does not guarantee accuracy or comprehensiveness.

IF YOU HAVE THE ATARI, WE HAVE THE PRINTERS.

GP-700AT
Full Color Printer — $599.

COLOR — is the big feature of our GP-700AT Atari direct-connect printer. It's fast, quiet and the bright colors will make any presentation come alive. We provide cable and screen dump programs to eliminate any installation problems. 850 interface is NOT required, even if you daisy-chain a disk drive or cassette recorder. You can print, plot and screen dump in 25 colors with Axiom's GP-700AT.

GP-550AT
Dual Mode Printer — $399.

QUALITY — is the keynote in our near letter quality mode with a density normally costing twice as much. You get friction, fully adjustable tractors, and even bottom feed for label printing. 850 interface is NOT required. We supply cable, manual and screen dump programs to get you off to a flying start. You can even daisy-chain a disk drive or cassette with the GP-100AT Atari direct-connect printer.

GP-100AT
Economical Printer — $299.

COST — is small but features are many. You get graphics and adjustable tractors. Uses full size standard fan-fold paper. Just take it home, plug it in and you'll be printing in minutes. The GP-100AT Atari direct-connect printer comes with screen dump to provide spectacular graphics, and you can daisy-chain other peripherals. Again, NO 850 interface is required.

AT-WORDSTORE
Printer Buffer — $299.

EFFICIENCY — is what you get with our 32K printer buffer. Features include pause and copy. WORDSTORE can be connected to Atari computers WITHOUT an 850 interface. Other peripherals can be daisy-chained. Accepts any Centronics-parallel input printer. Two year warranty. Best of all, WORDSTORE is priced well below other printer buffers.

AT-846
Expansion Interface — $99.

SIMPLICITY — it's just like Atari's 850 interface, but without the four funny serial ports. Great for interfacing any parallel printer to your Atari computer. Can daisy-chain other peripherals, too. By the way, if you think up a name for this product, drop us a line. If we use it, we'll send you a color printer free. Contest ends 12-31-83.

AXIOM®

AXIOM CORPORATION

1014 Griswold Avenue
San Fernando, CA 91340
(213) 365-9521 • TWX: 910-496-1746



LISTING CONVENTIONS

Table Information

Our custom font listings represent each ATASCII character as it appears on the video screen. You generate some characters by a single keystroke, for example, the regular alphabet. Others require a combination or sequence of keystrokes. In this table, ESC means *press and release* the escape key before pressing another key. CTRL or SHIFT means *press and hold* the control or shift key while simultaneously pressing the following key.

The Atari logo key ([logo]) "toggles" inverse video for all alphanumeric and punctuation characters. Press the logo key once to turn it on; press again to turn it off. In the XL line there is no logo key; inverse video is controlled by a key on the function row. Decimal values are given as reference, and correspond to the CHR$ values often used in BASIC listings.

NORMAL VIDEO

FOR THIS	TYPE THIS	DECIMAL VALUE
♥	CTRL ,	0
	CTRL A	1
	CTRL B	2
	CTRL C	3
	CTRL D	4
	CTRL E	5
	CTRL F	6
	CTRL G	7
	CTRL H	8
	CTRL I	9
	CTRL J	10
	CTRL K	11
	CTRL L	12
	CTRL M	13
	CTRL N	14
	CTRL O	15
	CTRL P	16
	CTRL Q	17
	CTRL R	18
	CTRL S	19
	CTRL T	20
	CTRL U	21
	CTRL V	22
	CTRL W	23
	CTRL X	24
	CTRL Y	25
	CTRL Z	26
	ESC ESC	27
↑	ESC CTRL −	28
↓	ESC CTRL =	29
←	ESC CTRL +	30
→	ESC CTRL *	31
	CTRL .	96
	CTRL ;	123
|	SHIFT =	124
	ESC SHIFT CLEAR	125
	ESC DELETE	126
	ESC TAB	127

INVERSE VIDEO

FOR THIS	TYPE THIS	DECIMAL VALUE
	[logo] CTRL ,	128
	[logo] CTRL A	129
	[logo] CTRL B	130
	[logo] CTRL C	131
	[logo] CTRL D	132
	[logo] CTRL E	133
	[logo] CTRL F	134
	[logo] CTRL G	135
	[logo] CTRL H	136
	[logo] CTRL I	137
	[logo] CTRL J	138
	[logo] CTRL K	139
	[logo] CTRL L	140
	[logo] CTRL M	141
	[logo] CTRL N	142
	[logo] CTRL O	143
	[logo] CTRL P	144
	[logo] CTRL Q	145
	[logo] CTRL R	146
	[logo] CTRL S	147
	[logo] CTRL T	148
	[logo] CTRL U	149
	[logo] CTRL V	150
	[logo] CTRL W	151
	[logo] CTRL X	152
	[logo] CTRL Y	153
	[logo] CTRL Z	154
	ESC SHIFT DELETE	156
	ESC SHIFT INSERT	157
	ESC CTRL TAB	158
	ESC SHIFT TAB	159
	[logo] CTRL .	224
	[logo] CTRL ;	251
	[logo] SHIFT =	252
	ESC CTRL 2	253
	ESC CTRL DELETE	254
	ESC CTRL INSERT	255

SHOPPER'S GUIDE

YOUR AD COULD APPEAR HERE CALL 415/661-3400

microscreens

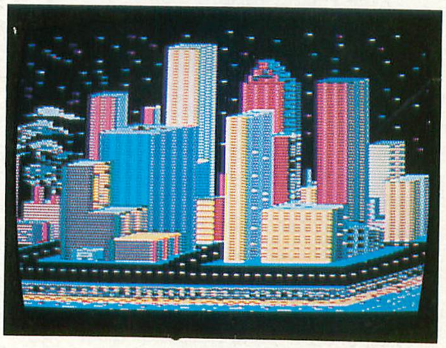
Make art with your Atari

You can create art with your Atari computer! Many techniques exist, including the use of commercial products such as **Micro-Painter, Graphic Master, PAINT, Fun With Art** and **Drawit**.* Or you might wish to try typing in **Keystroke Artist**, a graphics-utility program that appeared in the August 1983 issue of ANTIC.



©Walt Disney Productions





From the wonderful world of Disney and the Atari computer of Laurent Basset comes this charming image of Mickey Mouse as the "Sorcerer's Apprentice." Laurent used his knowledge of machine language and display list interrupts (along with Datasoft's **Micro-Painter**) to produce this microscreen. Currently a nineteen-year-old freshman at UCLA, Laurent is new to the U.S. (he immigrated from France four years ago) but not to computing (he purchased his Atari 800 three years ago). He is studying graphic design and visual communication, and has worked on Walt Disney's "TRON" and on several projects for Atari.

"Waterfall' by Pauline Murabayashi is a fine example of a microscreen created with the aid of Datasoft's **Graphic Master**. Its bright blue waters may have been inspired by those of her own state, Hawaii.

"Nightlife" is a sharp-eyed rendering by Michael Sharp of an unidentified American city at night. Michael hails from Coventry, Rhode Island; his microscreen could well be an idealized vision of Providence, Rhode Island's capital and hub of the state's industry and commerce.

*Micro-Painter and Graphic Master are products of DataSoft, 9421 Winnetka Ave., Chatsworth, CA 91311. PAINT is a product of Atari, Inc. Fun With Art is a product of Epyx, 1043 Kiel Court, Sunnyvale, CA 94089. Drawit is a product of APX (Atari Program Exchange), P.O. Box 3705, Santa Clara, CA 95055.


21
1
From the creators of
Frogger,™ Scramble,™ Super Cobra,™
Tutankham™ and Time Pilot™. . .
KONAMI'S
POOYAN
by DATASOFT®
Datasoft® brings you POOYAN,® the popular arcade game from Konami, for home computers. Enjoy all the fun, fast action, colorful, high-resolution graphics and arcade-style play without all the quarters.
Quick reflexes and fast thinking will protect your helpless piglets in the forest valley from a pack of vicious, hungry wolves who travel up and down the valley clinging to balloons and throwing objects at you. Shoot your arrows to burst their balloons or throw chunks of meat to get them to let go, but don't give up, because the wolves won't!
POOYAN® is a super challenge that will keep any experienced gamer glued to the screen. And Datasoft® will be making it available for Apple,® Atari® 400/800/1200, Radio Shack Color, Commodore 64 and VIC 20 computer systems.
Datasoft Inc.®
HOME COMPUTER SOFTWARE
9421 Winnetka Avenue
Chatsworth, California 91311
(213) 701-5161
©1983 Datasoft® Inc.
Datasoft® is a registered trademark of Datasoft Inc.
POOYAN® is a registered trademark of Konami® Industries Company Ltd.
Ross



AMDISK YOUR ATARI®!

The waiting is over! Amdisk III, a revolutionary 3″ microfloppy disk drive, is now Atari-compatible. Amdisk III provides superior performance vs. other Atari-compatible disk drives. An intelligent controller provides compatibility with the Atari 400, 800 and XL Series computers. Maximize the power of your Atari computer with the Amdisk III. See your local computer dealer and ask for the Amdisk III.

- 3″ cartridge media for convenience and durability.
- Controls up to four (4) disk drives, each programmable for single or double-density operation. In double-density operation, Amdisk III provides up to 360K bytes of on-line, formatted storage capacity (180K bytes/side x 2 drives).
- Double-sided cartridge allows user access to an additional 360K bytes by manually flipping it over. Total formatted storage is 720K bytes, 8 times the storage available on single-density 5¼″ diskettes.
- May be used in conjunction with any 5¼″ disk drive to boot Atari-compatible software.
- Fully-compatible with OS/A+ Version 4.
- Software Support: DataPerfect™, LetterPerfect™ and SpellPerfect™ from LJK Inc™. and OS/A+, operating system software from OSS available from Amdek on 3″ cartridge. Additional software packages of leading vendors will be available soon.

Atari® is a registered trademark of Atari, Inc.
DataPerfect™, LetterPerfect™, SpellPerfect™ and LJK Inc.™ are trademarks of LJK Inc.™

AMDEK

2201 Lively Blvd. • Elk Grove Village, IL 60007
(312) 364-1180 TLX: 25-4786

REGIONAL OFFICES: Southern Calif. (714) 662-3949 • Texas (817) 498-2334
Northern Calif. (408) 370-9370 • Denver (303) 794-1497